Docket No.: 60188-694 **PATENT**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of	:	Customer Number: 20277
Makoto FUJIWARA, et al.	:	Confirmation Number:
Serial No.:	:	Group Art Unit:
Filed: October 30, 2003	:	Examiner: Unknown

For: PROGRAM UPDATE METHOD AND SERVER

CLAIM OF PRIORITY AND
TRANSMITTAL OF CERTIFIED PRIORITY DOCUMENT

Mail Stop CPD
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

In accordance with the provisions of 35 U.S.C. 119, Applicants hereby claim the priority of:

Japanese Patent Application No. 2002-331992, filed November 15, 2002

cited in the Declaration of the present application. A certified copy is submitted herewith.

Respectfully submitted,

MCDERMOTT, WILL & EMERY

Michael E. Fogarty
Registration No. 36,139

600 13th Street, N.W.
Washington, DC 20005-3096
(202) 756-8000 MEF:tlb
Facsimile: (202) 756-8087
Date: October 30, 2003

60188-694
FUJIWARA et al.
October 30, 2003

日本国特許庁
JAPAN PATENT OFFICE

McDermott, Will & Emery

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: ２００２年１１月１５日

出願番号
Application Number: 特願２００２－３３１９９２

[ST. 10/C]: [ＪＰ２００２－３３１９９２]

出願人
Applicant(s): 松下電器産業株式会社

２００３年 ８月１２日

特許庁長官
Commissioner,
Japan Patent Office 今井康夫

出証番号 出証特２００３－３０６４８４５

【書類名】　特許願

【整理番号】　5037540113

【提出日】　平成14年11月15日

【あて先】　特許庁長官　殿

【国際特許分類】　G06F 15/00

【発明者】

【住所又は居所】　大阪府門真市大字門真１００６番地　松下電器産業株式会社内

【氏名】　藤原　睦

【発明者】

【住所又は居所】　大阪府門真市大字門真１００６番地　松下電器産業株式会社内

【氏名】　根本　祐輔

【発明者】

【住所又は居所】　大阪府門真市大字門真１００６番地　松下電器産業株式会社内

【氏名】　安井　純一

【発明者】

【住所又は居所】　大阪府門真市大字門真１００６番地　松下電器産業株式会社内

【氏名】　前田　卓治

【発明者】

【住所又は居所】　大阪府門真市大字門真１００６番地　松下電器産業株式会社内

【氏名】　伊藤　孝幸

【発明者】

【住所又は居所】　大阪府門真市大字門真１００６番地　松下電器産業株式会社内

【氏名】　山田　泰司

【発明者】

【住所又は居所】　大阪府門真市大字門真１００６番地　松下電器産業株式会社内

【氏名】　井上　信治

【特許出願人】

【識別番号】　000005821

【氏名又は名称】　松下電器産業株式会社

【代理人】

【識別番号】　100077931

【弁理士】

【氏名又は名称】　前田　弘

【選任した代理人】

【識別番号】　100094134

【弁理士】

【氏名又は名称】　小山　廣毅

【選任した代理人】

【識別番号】　100110939

【弁理士】

【氏名又は名称】　竹内　宏

【選任した代理人】

【識別番号】　100110940

【弁理士】

【氏名又は名称】　嶋田　高久

【選任した代理人】

【識別番号】　100113262

【弁理士】

【氏名又は名称】　竹内　祐二

【選任した代理人】

【識別番号】　100115059

【弁理士】

【氏名又は名称】　今江　克実

【選任した代理人】

【識別番号】　100115510

【弁理士】

【氏名又は名称】　手島　勝

【選任した代理人】

【識別番号】　100115691

【弁理士】

【氏名又は名称】　藤田　篤史

【手数料の表示】

【予納台帳番号】　014409

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【包括委任状番号】　0006010

【プルーフの要否】　要

【書類名】　明細書

【発明の名称】　プログラム更新方法およびサーバ

【特許請求の範囲】

【請求項１】　ＬＳＩと外部メモリとを含むシステムにおいて、前記外部メモリに格納され、前記ＬＳＩ固有の固有鍵で暗号化された固有鍵暗号化プログラムを更新する方法であって、

前記システムが、サーバから送信された，共有鍵で暗号化された共有鍵暗号化プログラムを受信する第１ステップと、

前記システムが、受信した前記共有鍵暗号化プログラムを復号することによって、平文プログラムを生成する第２ステップと、

前記システムが、前記平文プログラムを、前記固有鍵で再暗号化し、新たな固有鍵暗号化プログラムとして前記外部メモリに格納する第３ステップとを備えたことを特徴とするプログラム更新方法。

【請求項２】　請求項１記載のプログラム更新方法において、

前記サーバから送信された共有鍵情報を、受信するステップと、

受信した前記共有鍵情報を用いて、平文共有鍵を生成するステップとをさらに備え、

前記第２ステップにおいて、前記平文共有鍵を用いて、前記共有鍵暗号化プログラムを復号する

ことを特徴とするプログラム更新方法。

【請求項３】　請求項２記載のプログラム更新方法において、

前記共有鍵情報は、前記平文共有鍵を平文第１中間鍵で暗号化した暗号化共有鍵と、前記平文第１中間鍵を平文第２中間鍵で暗号化した暗号化第１中間鍵とを含むものである

ことを特徴とするプログラム更新方法。

【請求項４】　請求項１記載のプログラム更新方法において、

前記ＬＳＩは、固有鍵情報が格納された内部メモリを備えており、

前記システムは、その起動時に、前記内部メモリに格納された固有鍵情報を用いて、平文固有鍵を生成するものであり、

前記第３ステップにおいて、前記平文プログラムの再暗号化のために、前記平文固有鍵を用いる
ことを特徴とするプログラム更新方法。

【請求項５】　請求項４記載のプログラム更新方法において、

前記固有鍵情報は、前記平文固有鍵を平文第３中間鍵で暗号化した暗号化固有鍵と、前記平文第３中間鍵を平文第４中間鍵で暗号化した暗号化第２中間鍵とを含むものである
ことを特徴とするプログラム更新方法。

【請求項６】　請求項４記載のプログラム更新方法において、

生成された前記平文固有鍵は、前記ＬＳＩ内のレジスタに格納され、前記固有鍵暗号化プログラムを実行する際の、平文プログラムへの復号化のために、用いられる
ことを特徴とするプログラム更新方法。

【請求項７】　請求項１記載のプログラム更新方法において、

前記ＬＳＩは、ブートプログラムが格納されたブートＲＯＭを備えており、

前記外部メモリは、サーバとの送受信を実行するための取得プログラムが格納されており、

前記システムは、前記共有鍵暗号化プログラムの受信を、前記外部メモリに格納された取得プログラムによって実行し、受信後の更新処理を、前記ブートＲＯＭに格納されたブートプログラムによって制御する
ことを特徴とするプログラム更新方法。

【請求項８】　請求項１記載のプログラム更新方法において、

前記サーバから送信された、前記平文プログラムのハッシュ値を受信するステップを備え、

前記第２ステップにおいて、復号した平文プログラムについて、受信したハッシュ値を用いてハッシュ検証を行う
ことを特徴とするプログラム更新方法。

【請求項９】　ＬＳＩを含むシステムにおけるプログラム更新のために、動作するサーバであって、

前記システムから、前記ＬＳＩのＩＤと、更新対象プログラムの識別情報であるアプリＩＤとを受信する第１ステップと、

アプリＩＤとＬＳＩＩＤとの対応関係を示す第１のテーブルを参照して、前記更新対象プログラムを前記システムに送信するか否かを決定する第２ステップと、

前記第２ステップで送信すると決定したとき、前記システムへ、前記更新対象プログラムを共有鍵で暗号化した共有鍵暗号化プログラム、および前記共有鍵の基になる共有鍵情報を送信する第３ステップとを実行する
ことを特徴とするサーバ。

【請求項１０】　請求項９記載のサーバにおいて、

前記システムから、前記更新対象プログラムの実行に必要なアプリ固有情報を要求する信号を受信する第４ステップと、

アプリ固有情報の送信履歴とＬＳＩＩＤとの対応関係を示す第２のテーブルを参照して、前記第４のステップで要求されたアプリ固有情報を送信するか否かを決定する第５のステップとを実行する
ことを特徴とするサーバ。

【請求項１１】　請求項９記載のサーバにおいて、

前記共有鍵情報は、平文共有鍵を平文第１中間鍵で暗号化した暗号化共有鍵と、前記平文第１中間鍵を平文第２中間鍵で暗号化した暗号化第１中間鍵とを含むものである
ことを特徴とするサーバ。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、鍵実装されたシステムやこれに用いるＬＳＩにおいて、セキュリティを保ちつつ、プログラムを更新する技術に関する。

【０００２】

【従来の技術】

従来、ＬＳＩを動作させるプログラムを不正な処理から守るために、予め定ま

ったメーカー鍵で暗号化したプログラムをメモリに記憶しておき、復号化して実行することが行われている。しかしながら、このようなシステムでは共通のメーカー鍵で暗号化されたプログラムを実行するＬＳＩが大量にあるため、たとえ１個の製品からでもメーカー鍵の情報が不正に漏洩すると、大量の製品においてプログラムを改ざん可能になってしまい、したがって、セキュリティを高めることができないという課題がある。

【０００３】

この課題を解決するために、ＬＳＩを動作させるプログラムをＬＳＩ毎に固有の固有鍵で暗号化し、製品としては固有鍵で暗号化されたプログラムのみを実行可能とする手法がある（特願２００２－２１５０９６、特願２００２－２５８４８１参照）。この手法を用いることにより、１個の製品で鍵情報が不正に漏洩したとしても、その他の製品への影響はないので、セキュリティを高めることができる。また、この手法の前提として、鍵を二重に暗号化する手法がある（特願２００１－２８６８８１参照。）。

【０００４】

なお、ここで挙げた特許出願はいずれも未だ出願公開されておらず、このため、記載すべき先行技術文献情報はない。

【０００５】

【発明が解決しようとする課題】

一般に、製品となっているＬＳＩに実装されたプログラムのアップデート（更新）は、ＳＳＬ接続によって通信路を安全に確保し、平文プログラム、またはメーカー鍵で暗号化されたプログラムをサーバからＬＳＩへ送信することによって行われている。しかしながら、この手法では、通信路を不正にアクセスされると、大量の製品で実行可能なプログラムが不正に入手されてしまうので、プログラム更新におけるセキュリティを高めることができない。

【０００６】

これを解決するために、上述の手法を用いた場合、ＬＳＩは固有鍵で暗号化されたプログラムのみを実行するので、平文プログラムやメーカー鍵で暗号化されたプログラムを送信しても、そのままでは実行することができない。

【０００７】

また、ＬＳＩ毎に異なる鍵で暗号化されたプログラムをサーバ側で準備し、ＬＳＩ毎に鍵情報を管理した上で、サーバからＬＳＩへ、ＬＳＩ毎に異なる鍵で暗号化されたプログラムを送信する方法も考えられるが、この方法では、莫大な手間とコストを必要とするため、現実的ではない。

【０００８】

前記の問題に鑑み、本発明は、ＬＳＩ固有の固有鍵で暗号化されたプログラムを実行可能なＬＳＩについて、高いセキュリティを保ちつつ、プログラムを更新する方法を提供することを課題とする。

【０００９】

【課題を解決するための手段】

前記の課題を解決するために、本発明が講じた解決手段は、ＬＳＩと外部メモリとを含むシステムにおいて、前記外部メモリに格納され、前記ＬＳＩ固有の固有鍵で暗号化された固有鍵暗号化プログラムを更新する方法として、前記システムが、サーバから送信された共有鍵で暗号化された共有鍵暗号化プログラムを受信する第１ステップと、前記システムが、受信した前記共有鍵暗号化プログラムを復号することによって平文プログラムを生成する第２ステップと、前記システムが、前記平文プログラムを前記固有鍵で再暗号化し、新たな固有鍵暗号化プログラムとして前記外部メモリに格納する第３ステップとを備えたものである。

【００１０】

前記本発明に係るプログラム更新方法において、前記サーバから送信された共有鍵情報を受信するステップと、受信した前記共有鍵情報を用いて平文共有鍵を生成するステップとをさらに備え、前記第２ステップにおいて、前記平文共有鍵を用いて前記共有鍵暗号化プログラムを復号するのが好ましい。

【００１１】

さらに、前記共有鍵情報は、前記平文共有鍵を平文第１中間鍵で暗号化した暗号化共有鍵と、前記平文第１中間鍵を平文第２中間鍵で暗号化した暗号化第１中間鍵とを含むのが好ましい。

【００１２】

また、前記本発明に係るプログラム更新方法において、前記ＬＳＩは、固有鍵情報が格納された内部メモリを備えており、前記システムは、その起動時に、前記内部メモリに格納された固有鍵情報を用いて平文固有鍵を生成するものとし、前記第３ステップにおいて、前記平文プログラムの再暗号化のために前記平文固有鍵を用いるものとする。

【００１３】

さらに、前記固有鍵情報は、前記平文固有鍵を平文第３中間鍵で暗号化した暗号化固有鍵と、前記平文第３中間鍵を平文第４中間鍵で暗号化した暗号化第２中間鍵とを含むのが好ましい。あるいは、生成された前記平文固有鍵は、前記ＬＳＩ内のレジスタに格納され、前記固有鍵暗号化プログラムを実行する際の平文プログラムへの復号化のために用いられるのが好ましい。

【００１４】

また、前記本発明に係るプログラム更新方法において、前記ＬＳＩは、ブートプログラムが格納されたブートＲＯＭを備えており、前記外部メモリは、サーバとの送受信を実行するための取得プログラムが格納されており、前記システムは、前記共有鍵暗号化プログラムの受信を、前記外部メモリに格納された取得プログラムによって実行し、受信後の更新処理を、前記ブートＲＯＭに格納されたブートプログラムによって制御するのが好ましい。

【００１５】

また、前記本発明に係るプログラム更新方法において、前記サーバから送信された前記平文プログラムのハッシュ値を受信するステップを備え、前記第２ステップにおいて、復号した平文プログラムについて、受信したハッシュ値を用いてハッシュ検証を行うのが好ましい。

【００１６】

また、本発明が講じた解決手段は、ＬＳＩを含むシステムにおけるプログラム更新のために動作するサーバとして、前記システムから前記ＬＳＩのＩＤと更新対象プログラムの識別情報であるアプリＩＤとを受信する第１ステップと、アプリＩＤとＬＳＩＩＤとの対応関係を示す第１のテーブルを参照して、前記更新対象プログラムを前記システムに送信するか否かを決定する第２ステップと、前記

第２ステップで送信すると決定したとき、前記システムへ、前記更新対象プログラムを共有鍵で暗号化した共有鍵暗号化プログラム、および前記共有鍵の基になる共有鍵情報を送信する第３ステップとを実行するものである。

【００１７】

そして、前記本発明に係るサーバにおいて、前記システムから、前記更新対象プログラムの実行に必要なアプリ固有情報を要求する信号を受信する第４ステップと、アプリ固有情報の送信履歴とＬＳＩＩＤとの対応関係を示す第２のテーブルを参照して、前記第４のステップで要求されたアプリ固有情報を送信するか否かを決定する第５のステップとを実行するのが好ましい。

【００１８】

また、前記本発明に係るサーバにおいて、前記共有鍵情報は、平文共有鍵を平文第１中間鍵で暗号化した暗号化共有鍵と、前記平文第１中間鍵を平文第２中間鍵で暗号化した暗号化第１中間鍵とを含むのが好ましい。

【００１９】

【発明の実施の形態】

以下、本発明の実施の形態について、図面を参照して説明する。なお、以下の説明では、Ｘ（鍵またはプログラム）を鍵Ｙを用いて暗号化して得た、暗号化された鍵またはプログラムのことを、Ｅｎｃ（Ｘ，Ｙ）と表すものとする。

【００２０】

図１は本実施形態に係る半導体装置としてのセキュアＬＳＩの内部構成を示すブロック図である。図１において、セキュアＬＳＩ１は外部バス１２０を介して、外部メモリ（フラッシュメモリ）１００や外部ツール１１０などと接続可能に構成されている。また、モードＩＤを与えることによって、その動作モードを設定することが可能になっている。

【００２１】

本実施形態に関わる主な構成要素について、簡単に説明する。

【００２２】

まず、セキュアＬＳＩ１は、書き換え不可領域１１を含むセキュアメモリ（セキュアＦｌａｓｈ）１０を備えている。この書き換え不可領域１１には、書き換

え不可領域書き込みフラグ１２が設けられている。書き換え不可領域書き込みフラグ１２は、モードＩＤが一度セキュアメモリ１０に書き込まれると、そのフラグ値が“可”から“済”になり、それ以降の書き換え不可領域への書き込みが不能になる。なお、本実施形態では、セキュアメモリ１０および外部メモリ１００はフラッシュメモリによって構成されているが、もちろんこれに限定されるものではなく、不揮発性のメモリであればどのようなものであってもかまわない。

【００２３】

また、暗号化部２はプログラムの暗号化や復号化を行うものであり、秘密鍵演算処理部２０と、鍵生成・更新シーケンサ３０とを備えている。秘密鍵演算処理部２０は各種の鍵、およびプログラム暗号種を格納するレジスタ（プログラム共有鍵格納レジスタ２１、プログラム固有鍵格納レジスタ２２、暗号鍵格納レジスタ２３等）を備えており、プログラムの暗号化処理又は復号化処理を含む複数のシーケンスを実行可能である。鍵生成・更新シーケンサ３０は秘密鍵演算処理部２０が実行可能な各シーケンスについて実行の諾否を判断し、実行が許されないと判断したシーケンスについて秘密鍵演算処理部２０の動作を禁止する。鍵生成更新シーケンサ３０はモードＩＤ格納レジスタ３１を有しており、このモードＩＤ格納レジスタ３１に格納されているモードＩＤに応じて、各シーケンスの実行の諾否を判断する。また、鍵またはプログラムがどのようなアルゴリズムや鍵長で暗号化されているかを示す暗号種別識別子を格納する暗号種別識別子格納レジスタ３２、およびプログラム暗号種を記憶する記憶部３３を備えている。

【００２４】

モードシーケンサ４０も、モードＩＤ格納レジスタ４１を備えており、モードＩＤ格納レジスタ４１に格納されているモードＩＤと、ジャンパ４３の値に応じて、外部ホストインターフェース（Ｉ／Ｆ）５０の動作、すなわち、外部メモリ１００に格納されたプログラムやデータをどのＩ／Ｆを介して読み込むか、を制御する。これにより、外部メモリ１００に格納された平文プログラムが実行できるか否かを制御することができる。さらに、モードシーケンサ４０は、鍵がどの手法によって暗号化されているかを示す暗号種別識別子を格納する暗号種別識別子格納レジスタ４２を備えている。

【００２５】

外部ホストＩ／Ｆ５０は、モードシーケンサ４０の制御に従って、プログラム処理部５１が有する暗号化用スルー部５２、実行用スルー部５３およびプログラム復号用暗号エンジン５４、並びに、データ処理部５５が有するスルー部５６およびコンテンツ暗号・復号用暗号エンジン５７のうちのいずれかを介して、外部メモリ１００や外部ツール１１０との間でプログラムやデータの入出力を行う。またプログラム復号用暗号エンジン５４は、プログラムの復号に用いるプログラム固有鍵を格納するためのプログラム固有鍵可能レジスタ５８を備えている。

【００２６】

ここで、後述する鍵生成モードと商品動作モードにおいては、実行用スルー部５３を介してプログラムを取り込むことが出来ないように構成されている。すなわち、セキュアＬＳＩ１は後述する鍵生成モードと商品動作モードにおいては、固有鍵で暗号化されたプログラム以外へは動作を遷移しないように構成されている。

【００２７】

ブートＲＯＭ６０は、セキュアＬＳＩ１の起動動作を制御するブートプログラムを格納している。ＨＡＳＨ演算部７０は、セキュアＬＳＩ１に読み込まれたプログラムについてその正当性を検証するために、ＨＡＳＨ値を演算する。

【００２８】

また、外部メモリ１００には、プログラムやコンテンツが格納されている。外部ツール１１０には、セキュアＬＳＩ１の最初の起動時にセキュアメモリ１０に格納する各種の初期値が格納されている。この初期値の種類は、設定される動作モードに応じて、異なったものになる。

【００２９】

図２は図１のセキュアＬＳＩ１を用いた開発および製品化の全体の流れを表す図である。図２に示すように、セキュアＬＳＩ１は、アドミニストレータモード（モードＩＤ：００）、鍵生成モード（モードＩＤ：０１）、開発モード（モードＩＤ：１０）および商品動作モード（モードＩＤ：１１）の４種類の動作モードで、動作する。

【００３０】

まず、アドミニストレータモードに設定されたセキュアＬＳＩ１は、管理者用ＬＳＩとして、動作する。管理者用ＬＳＩでは、鍵生成プログラムが開発され（ＰＡ１）、また、その鍵生成プログラムが任意の鍵生成鍵を用いて暗号化される（ＰＡ２）。

【００３１】

鍵生成モードに設定されたセキュアＬＳＩ１は、鍵生成用ＬＳＩとして、動作する。鍵生成用ＬＳＩでは、管理者用ＬＳＩにおいて生成された、暗号化された鍵生成プログラムが実装され（ＰＢ１）、この鍵生成プログラムを実行することによって、各種の鍵が生成される（ＰＢ２）。

【００３２】

開発モードに設定されたセキュアＬＳＩ１は、開発用ＬＳＩとして、動作する。開発用ＬＳＩでは、実際の製品で実行されるアプリケーション用プログラムが開発される（ＰＣ１）。そして、このアプリケーション用プログラムが、プログラム共有鍵を用いて暗号化される（ＰＣ２）。

【００３３】

商品動作モードに設定されたセキュアＬＳＩ１は、実際の商品ＬＳＩとして、動作する。商品ＬＳＩでは、開発用ＬＳＩにおいて生成された、プログラム共有鍵で暗号化されたアプリケーション用プログラムが実装され、その内部で、プログラム固有鍵で暗号化されたアプリケーション用プログラムに、変換される（ＰＤ１）。その後、通常の商品ＬＳＩとして動作する（ＰＤ２）。なお、この変換処理は、開発用ＬＳＩでも、アプリケーション用プログラムのデバッグのために、実行可能になっている（ＰＣ３）。

【００３４】

以下、上記のように構成されたセキュアＬＳＩ１の商品動作モードにおける通常動作およびセキュアアップデート動作の詳細について、フローチャートおよびデータフローを参照して、説明する。

【００３５】

図３はブートプログラムの全体的な処理の流れを示すフローチャートである。

セキュアＬＳＩ１に電源が投入されると、ブートＲＯＭ６０に格納されたブートプログラムがＣＰＵ６５によって実行される。図３に示すように、まず、各ハードウェアを初期化する（ＳＺ０）。そして、外部ツール１１０からさまざまな初期値を読み込み、セキュアメモリ１０に設定する（ＳＺ１）。

【００３６】

図４は初期値設定処理ＳＺ１のフローチャートである。まず、ジャンパ４４で、セキュアメモリ１０がＬＳＩ内に実装されているか否かの判定を行う（ＳＺ１１）。次に、書き換え不可領域書き込みフラグ１２が“済”であるか否かを判定し（ＳＺ１２）、“済”であるときは（Ｙｅｓ）すでにセキュアメモリ１０に初期値が設定されているので、処理ＳＺ１を終了する。書き換え不可領域書き込みフラグ１２が“可”であるときは（Ｎｏ）、セキュアメモリ１０に初期値を書き込んでいく（ＳＺ１３～ＳＺ１８）。モードＩＤに加えて、暗号化されたプログラム固有鍵、アドレス管理情報、データ固有鍵をセキュアメモリ１０の書き換え不可領域１１に書き込む。なお、最初の判定の結果、セキュアメモリ１０がＬＳＩの外部にあると判定されたときは（ＳＺ１４でＮｏ）、モードＩＤは商品動作モードを表す値に上書きされる（ＳＺ１５）。これにより、セキュアメモリ１０がＬＳＩパッケージ外にあるような不正な製品は、商品動作モードでしか動作できない。

【００３７】

次に、書き込み不可領域書き込みフラグ１２を“済”にセットする（ＳＺ１９）。これによって、以後の書き換え不可領域１１の書き換えはできなくなる。さらに、通常領域１３，１４に暗号種別識別子および実装モードフラグを書き込む（ＳＺ１Ａ）。そして、モードＩＤがアドミニストレータモード以外のモードを示すときは（ＳＺ１ＢでＮｏ）、これらに加えて、暗号化された共有鍵／鍵生成鍵も通常領域１３，１４に書き込む（ＳＺ１Ｃ）。

【００３８】

その後、前処理ＳＺ２を実行する。ここでは、セキュアメモリ１０の書き込み不可領域１１に設定されたモードＩＤが、鍵生成・更新シーケンサ３０のモードＩＤ格納レジスタ３１と、モードシーケンサ４０のモードＩＤ格納レジスタ４１

とに設定される。また、セキュアメモリ１０の第１の通常領域１３に設定された暗号種別識別子が、鍵生成・更新シーケンサ３０の暗号種別識別子格納レジスタ３２と、モードシーケンサ４０の暗号種別識別子格納レジスタ４２とに設定される。さらに、セキュアメモリ１０の書き換え不可領域１１に格納されたアドレス管理情報が、ＭＥＭＣ８０の暗号アドレス区分格納レジスタ８１に設定される。ここまでの動作は、図２における初期値設定フェーズＰＡ０，ＰＢ０，ＰＣ０，ＰＤ０に対応している。

【００３９】

その後は、モードＩＤの値に応じて、それぞれのモードにおける動作を行う（ＳＺ３）。このようにモードＩＤの値に応じて、セキュアＬＳＩで行う動作を制限することによりプログラムの秘匿性を高めている。

【００４０】

次に通常の商品動作（通常ブート処理）について詳細に説明する。

【００４１】

モードＩＤが「１１」のとき、セキュアＬＳＩ１は商品動作モードになり、実装モードフラグの値に応じて（ＳＤ０）、プログラム実装処理ＳＤ１、または通常ブート処理ＳＤ２を実行する。

【００４２】

図５はプログラム実装処理ＳＤ１のフローチャート、図６、７はデータフローである。プログラム実装処理ＳＤ１においては、セキュアメモリ１０に格納された固有鍵情報を用いてプログラム固有鍵を（ＳＤ１１、ＳＤ１２）、共有鍵情報を用いてプログラム共有鍵を復号し（ＳＤ１３、ＳＤ１４）、復号されたプログラム共有鍵とプログラム固有鍵を用いて外部メモリ１００に格納されたプログラムＥｎｃ（プログラム、プログラム共有鍵）をＥｎｃ（プログラム、プログラム固有鍵）へと変換する（ＳＤ１５－ＳＤ１７）。その後、プログラムの正当性の検証を行い（ＳＤ１８）、正当であるならば実装モードフラグをＯＦＦに設定する（ＳＤ１９）。これにより次回の起動時からはプログラム実装処理ＳＤ１は行われない。そして最後にセキュアメモリ１０に格納されたプログラム共有鍵と、外部メモリ１００に格納されたプログラムＥｎｃ（プログラム、プログラム共有

鍵）は削除される（ＳＤ１Ａ、ＳＤ１Ｂ）。

【００４３】

図８は通常ブート処理ＳＤ２のフローチャート、図９、１０はデータフローである。通常ブート処理ＳＤ２においては、まず、内部メモリとしてのセキュアメモリ１０の書き込み不可領域１１に格納された、固有鍵鍵情報としての暗号化されたプログラム固有鍵、すなわち、暗号化固有鍵Ｅｎｃ（プログラム固有鍵（平文）、ＭＫ０（平文第３中間鍵））および暗号化第２中間鍵Ｅｎｃ（ＭＫ０，ＣＫ０（平文第４中間鍵））を秘密鍵演算処理部２０の暗号鍵格納レジスタに設定する（ＳＤ２１）。そして、この暗号化されたプログラム固有鍵を、鍵生成・更新シーケンサ３０に実装されたプログラム暗号種を用いて復号し、プログラム固有鍵を得る（ＳＤ２２）。得られたプログラム固有鍵は秘密鍵演算処理部２０のプログラム固有鍵格納レジスタ２２と、外部ホストＩ／Ｆ５０のプログラム復号用暗号エンジン５４のプログラム固有鍵格納レジスタ５８に設定する（ＳＤ２３）。

【００４４】

その後、セキュアメモリ１０の書き込み不可領域１１に格納されているデータ固有ＩＤを秘密鍵演算処理部２０の固有ＩＤ格納レジスタに設定する（ＳＤ２４）。また、ＣＰＵ６５によって乱数を生成し、秘密鍵演算処理部２０の乱数格納レジスタに設定する（ＳＤ２５）。そして、秘密鍵演算処理部２０によって、データ固有ＩＤと乱数からデータ固有鍵を生成する（ＳＤ２６）。コンテンツの再生にはデータ固有鍵を用いて行う。データ固有鍵は乱数を用いて生成されるため、起動毎に異なることとなり、コンテンツ再生の安全性が高まる。

【００４５】

その後、外部メモリ１００に格納されていた，プログラム固有鍵で暗号化されたプログラムＥｎｃ（プログラム，プログラム固有鍵）を、外部ホストＩ／Ｆ５０が有するプログラム処理部５１のプログラム復号用暗号エンジン５４を介して復号し、ＨＡＳＨ演算部７０に取り込み、ＨＡＳＨ値を演算する（ＳＤ２７）。復号に用いられる鍵としては外部ホストＩ／Ｆのプログラム固有鍵格納レジスタ５８に格納されたプログラム固有鍵が用いられる。そして、この演算したＨＡＳ

Ｈ値と、セキュアメモリ１０の通常領域１３に格納されていたＨＡＳＨ値とを比較し、プログラムが改ざんされていないかどうかをチェックする（ＳＤ２８）。ＨＡＳＨ値が一致していたとき（ＳＤ２９でＮｏ）、外部メモリ１００に格納されていたプログラムＥｎｃ（プログラム，プログラム固有鍵）に処理を遷移し、アプリケーションを実行する（ＳＤ２Ａ）。一方、ＨＡＳＨ値が一致していないとき（ＳＤ２９でＹｅｓ）は、何らかの不正が行われたものと推定して、不正アクセス時制御による処理を実行する（ＳＤ２Ｂ）。

【００４６】

ここで、上述のように商品として動作するセキュアＬＳＩについて、プログラムの更新を行う処理について、図面を参照しながら説明する。図１１はプログラム更新の際に、サーバ３とセキュアＬＳＩ１を含むシステムとの間におけるデータのやり取りを示すフローである。

【００４７】

図１１に示すように、まず、セキュアＬＳＩ１がプログラム更新処理を起動すると、サーバ３はシステムからセキュアＬＳＩ１のＩＤを受信してＩＤ認証を行い、認証した場合は、セキュアＬＳＩ１とＳＳＬ接続する（ＵＤ１）。これにより、サーバ３とセキュアＬＳＩ１を含むシステムとの間の通信路の安全性が一応確保される。

【００４８】

通信路が確保されると、システムは、更新対象プログラムの識別情報であるアプリＩＤをサーバ３へ送信する（ＵＤ２）。サーバ３は、更新可能なプログラムのアプリＩＤと、プログラムを動作させて良いＬＳＩのＩＤとの対応関係を示す第１のテーブル４を管理しており、この第１のテーブル４に基づいて、プログラムを送信してよいか否かを判断する。セキュアＬＳＩ１のＩＤと、更新を要求されたプログラムのアプリＩＤとの対応が確認されると、サーバ３は、この更新対象プログラムの送信を開始する。

【００４９】

まず、サーバ３から、更新対象プログラムの付加情報をセキュアＬＳＩ１へ送信する（ＵＤ３）。ここでの付加情報は、セキュアＬＳＩ１側でプログラムを更

新してよいか否かを認証させるための署名、更新対象プログラムのサイズ、および、更新対象プログラムのハッシュ値（平文での値）等を含む。セキュアＬＳＩ１は、付加情報として送信された署名を用いて認証を行い、また、送信されたプログラムサイズを基にして、外部メモリ１００に更新可能な空き領域があるか否かを判断する。そして、更新が可能であると判断した場合は、サーバ３に対して、共有鍵情報を送信するよう要求する（ＵＤ４）。

【００５０】

サーバ３は、要求を受信すると、共有鍵情報として、暗号化共有鍵Ｅｎｃ（プログラム共有鍵（平文），ＭＫ１（平文第１中間鍵））および暗号化第１中間鍵Ｅｎｃ（ＭＫ１，ＣＫ１（平文第２中間鍵））をセキュアＬＳＩ１へ送信する（ＵＤ５）。セキュアＬＳＩ１は、共有鍵情報を用いてプログラム共有鍵を復号し、復号された状態でハッシュ演算を行い、正当性の検証をする。プログラム共有鍵を正常に復号すると、システムはサーバ３に対して、共有鍵暗号化プログラムを送信するよう要求する（ＵＤ６）。サーバ３は、要求を受信すると、プログラムＥｎｃ（プログラム、プログラム共有鍵）をシステムへ送信する（ＵＤ７）。セキュアＬＳＩ１は、Ｅｎｃ（プログラム、プログラム共有鍵）をＥｎｃ（プログラム、プログラム固有鍵）へと変換する。さらに、変換されたＥｎｃ（プログラム、プログラム固有鍵）を平文プログラムに復号し、ハッシュ演算して、先に付加情報として受信したハッシュ値との比較によって、正当性を検証する。ここでの処理については、後で詳細に説明する。

【００５１】

共有鍵暗号化プログラムが固有鍵暗号化プログラムに正常に変換できたとき、セキュアＬＳＩ１を含むシステムはサーバ３へ、アプリ固有情報を送信するよう要求する（ＵＤ８）。アプリ固有情報はプログラムの実行に必要な情報を含んでおり、アプリ固有情報がないと、セキュアＬＳＩ１は更新したプログラムを実行できない。サーバ３は、アプリ固有情報の送信履歴とＬＳＩのＩＤとの対応関係を示す第２のテーブル５も管理しており、同一のセキュアＬＳＩには複数のアプリ固有情報を送信しないようにする。よって、同一のセキュアＬＳＩは複数回、同一のプログラムを更新することができない。

【００５２】

サーバ３は、アプリ固有情報を送信してよいと判断したとき、アプリ固有情報をセキュアＬＳＩ１を含むシステムへ送信し（ＵＤ９）、セキュアＬＳＩ１がこれをハッシュ演算して正当性を検証すると、プログラム更新が終了し、通信は切断される（ＵＤ１０）。

【００５３】

なお、本発明におけるサーバ３とセキュアＬＳＩを含むシステムとの間におけるデータのやり取りは上述のフローに限定されるものではない。例えば、サーバ３は、必ずしも、第２のテーブルを管理し、同一のセキュアＬＳＩに複数のアプリ固有情報を送信しないようにする必要はない。しかしながら、同一のセキュアＬＳＩに複数回、同一のプログラムを送信しないようにすることにより、プログラムの秘匿性はより高まる。

【００５４】

また、付加情報、共有鍵情報、共有鍵暗号化プログラムは、必ずしも、サーバ３からセキュアＬＳＩへと別々に送信する必要はなく、その一部、または全部をまとめたプログラムパッケージとして一度に送信してもよい。

【００５５】

セキュアＬＳＩ１における、共有鍵暗号化プログラムＥｎｃ（プログラム、プログラム共有鍵）から固有鍵暗号化プログラムＥｎｃ（プログラム、プログラム固有鍵）への変換について、図面を参照しながら詳述する。図１２は外部メモリ１００に格納された、プログラム更新に係わるプログラムの構成を示す図である。

【００５６】

図１２に示すように、外部メモリ１００には、固有鍵でそれぞれ暗号化された、暗号化制御プログラム２００（Ｅｎｃ（制御プログラム、プログラム固有鍵））、および暗号化アプリケーションプログラム２１０（Ｅｎｃ（アプリケーションプログラム、プログラム固有鍵））が格納されている。

【００５７】

暗号化制御プログラム２００はアプリケーション起動部２０１およびプログラ

ム更新制御部２０５を備え、プログラム更新制御部２０５は共有鍵復号部２０６、プログラム固有鍵暗号化処理部２０７およびプログラム更新成否判定部２０８を備えている。

【００５８】

アプリケーション起動部２０１はブートＲＯＭ６０に格納されたブートプログラムからの指示を受けて、暗号化アプリケーションプログラム２１０を起動する。共有鍵復号部２０６はサーバ３から送信された共有鍵情報を基にして、鍵生成・更新シーケンサ３０を用いて、プログラム共有鍵を復号する。プログラム固有鍵暗号化処理部２０７は鍵生成・更新シーケンサ３０を用いて、共有鍵暗号化プログラムＥｎｃ（プログラム、プログラム共有鍵）から固有鍵暗号化プログラムＥｎｃ（プログラム、プログラム固有鍵）への変換を行う。プログラム更新成否判定部２０８は、固有鍵暗号化プログラムＥｎｃ（プログラム、プログラム固有鍵）を平文プログラムに復号し、ハッシュ検証によってプログラム更新の成否を判定する。そして、プログラム更新が成功したときは、古いプログラムを削除し、プログラム格納先・サイズなどの情報をセキュアメモリ１０に格納する。

【００５９】

暗号化アプリケーションプログラム２１０は、通常のアプリケーションプログラムである通常動作部２１１の他、サーバや記録媒体から新しいアプリケーションプログラムを取得するためのプログラム取得部２１２もアプリケーションプログラムとして備えている。プログラムの更新は、これらのような外部メモリ１００に格納されたプログラムを用いて行われる。

【００６０】

図１３は共有鍵暗号化プログラムから固有鍵暗号化プログラムへの変換処理を含む、プログラム更新処理を示すフローチャートである。

【００６１】

アプリケーションプログラムの実行中（ＳＸ１）に、ユーザの操作等の外部要因によってプログラムの更新が要求されると、システムはこれを検知し、プログラム更新を開始するために、通常動作部２１１が暗号化アプリケーションプログラム２１０の取得プログラムとしてのプログラム取得部２１２を起動する（ＳＸ

２）。

【００６２】

プログラム取得部２１２はサーバ３と通信して、認証や、共有鍵情報・プログラムの取得を行う（ＳＸ３）。サーバ３から共有鍵情報を取得すると、共有鍵復号部２０６はプログラム共有鍵を復号する（ＳＸ４，ＳＸ５）。すなわち、共有鍵鍵情報としての暗号化されたプログラム共有鍵Ｅｎｃ（プログラム共有鍵，ＭＫ２）、Ｅｎｃ（ＭＫ２，ＣＫ）を秘密鍵演算処理部２０の暗号鍵格納レジスタ２３に設定し、この暗号化されたプログラム共有鍵を、鍵生成・更新シーケンサ３０に実装されたプログラム暗号種を用いて復号し、プログラム共有鍵を得る。得られたプログラム共有鍵は秘密鍵演算処理部２０のプログラム共有鍵格納レジスタ２１に格納される。

【００６３】

次に、プログラム固有鍵暗号化処理部２０７が、共有鍵暗号化プログラムから固有鍵暗号化プログラムへの変換を行う。すなわち、サーバ３から送信され外部メモリ１００に格納されていたプログラムＥｎｃ（プログラム，プログラム共有鍵）を、外部ホストＩ／Ｆ５０が有するプログラム処理部５１の暗号化用スルー部５２を介して、秘密鍵演算処理部２０に取り込む（ＳＸ6'）。そして、取り込んだプログラムを、プログラム共有鍵格納レジスタ２１に格納されたプログラム共有鍵で復号した後、プログラム固有鍵格納レジスタ２２に格納されたプログラム固有鍵で暗号化し、プログラムＥｎｃ（プログラム，プログラム固有鍵）を得る。なお、上述したように、プログラム固有鍵はシステムの起動時にすでに復号されており、秘密鍵演算処理部２０のプログラム固有鍵格納レジスタ２２に格納されている。

【００６４】

最後に、プログラム更新成否判定部２０８が、プログラム更新の成否を判定する。すなわち、Ｅｎｃ（プログラム，プログラム固有鍵）を外部メモリ１００に書き込んだ（ＳＸ８）後、外部ホストＩ／Ｆ５０が有するプログラム処理部５１のプログラム復号用暗号エンジン５３を用いて、復号して取り込み（ＳＸ９）、平文状態でのハッシュ値を演算する（ＳＸ１０）。演算されたハッシュ値は、プ

ログラム取得部２１２が暗号化プログラムとともに取得したハッシュ値と比較され、この比較によって更新の成否が判定される（ＳＸ１１）。更新が成功したときは、古いプログラムを消去する（ＳＸ１２）一方、更新が失敗したときは、送信されたプログラムを消去する（ＳＸ１３）。そしてプログラム格納先、サイズなどの情報をセキュアメモリ１０に書き込み（ＳＸ１４）、更新処理が終了する。

【００６５】

上述のプログラム更新方法を用いることによって、サーバからプログラム共有鍵暗号化プログラムを送信すると、セキュアＬＳＩにおいて、暗号化する鍵がプログラム共有鍵からプログラム固有鍵に変換されて、システムに実装される。このため、たとえサーバからセキュアＬＳＩへの通信路が不正にアクセスされプログラム共有鍵暗号化プログラムが盗み出されたとしても、このプログラムによってセキュアＬＳＩを動作させることは、できない。また更新の結果、ユーザの持つ各製品では、互いに異なる固有鍵によって暗号化されたプログラムが実装されることになり、秘匿性が向上する。また、万一、暗号を破られた場合でも、被害を受ける製品の数が限定されることになり、従来よりもセキュリティが高まる。

【００６６】

なお、本実施形態において、共有鍵情報をサーバから取得しているが、これは、商品動作モード「１１」におけるプログラム実装（ＳＤ１）の最後において、復号されたプログラム共有鍵およびセキュアメモリ１０上の共有鍵情報を削除しているためであり、これらを削除しない場合は、共有鍵情報はサーバから取得する必要はなく、セキュアメモリ１０から読み出して復号すればよい。

【００６７】

また、本実施形態では、外部要因によってプログラムの更新の開始が指示され、通常瞳サブ２１１がプログラム取得部２１２を起動し、プログラムの取得後はブートプログラムによって各処理を指示するものとしたが、本発明はこれに限られるものではない。例えば、ブートプログラムがプログラム取得部２１２を起動する構成にすることによって、さらにセキュリティを高めることができる。

【００６８】

また、プログラム固有鍵は必ずしも製品毎に固有である必要はなく、品種ごとまたは複数個毎に同一であっても良い。１個の製品において暗号を破られたときに被害を少なくするのが本願発明のねらいであり、同一の鍵で暗号化されたプログラムを有するＬＳＩの数を少しでも減らすだけで効果は充分に発揮できる。さらに言えば、プログラム固有鍵は全て共通であったとしても、通信路が破られ共有鍵暗号化プログラムが盗み出されたとしてもそのままではセキュアＬＳＩで動作できないので、鍵を共有鍵から固有鍵に書き換えるだけでも、効果は発揮できる。

【００６９】

【発明の効果】

以上のように本発明によると、プログラムをＬＳＩ毎の固有鍵で再暗号化して実行する秘匿性の高いセキュアＬＳＩにおいても、サーバから同一のプログラムを送信するだけで、プログラムの更新をすること可能となる。

【００７０】

また、サーバからセキュアＬＳＩへの通信路が不正にアクセスされて共有鍵暗号化プログラムが盗み出されたとしても、そのプログラムではセキュアＬＳＩを動作させることができないので、秘匿性が向上する。また、万一、暗号を破られた場合でも、被害を受ける製品の数が限定されることになり、従来よりもセキュリティが高まる。

【００７１】

さらに、サーバから受信した共有鍵やプログラムの正当性を平文状態のハッシュ値を用いて行うので、通信路における暗号化状態でのハッシュ値で行うよりもハッシュ値の改ざんが行いにくくなり、セキュリティは高まる。

【図面の簡単な説明】

【図１】

本発明の実施形態に係るセキュアＬＳＩの構成を示すブロック図である。

【図２】

図１のセキュアＬＳＩを用いた開発および製品化の全体の流れを表す図である。

【図３】

ブートプログラムの全体的な処理の流れを示すフローチャートである。

【図４】

セキュアメモリ初期値設定ＳＺ１のデータフローである。

【図５】

商品動作モードにおけるプログラム実装処理ＳＤ１のフローチャートである。

【図６】

プログラム実装処理ＳＤ１のデータフロー１である。

【図７】

プログラム実装処理ＳＤ１のデータフロー２である。

【図８】

商品動作モードにおける通常ブート処理ＳＤ２のフローチャートである。

【図９】

通常ブート処理ＳＤ２のデータフロー１である。

【図１０】

通常ブート処理ＳＤ２のデータフロー１である。

【図１１】

プログラム更新におけるサーバとの通信を示すフローチャートである。

【図１２】

外部メモリ１００に格納された，プログラム更新に係わるプログラムの構成を示す図である。

【図１３】

プログラムの更新処理を示すフローチャートである。

【符号の説明】

１　セキュアＬＳＩ

３　サーバ

４　第１のテーブル

５　第２のテーブル

１０　セキュアメモリ（内部メモリ）

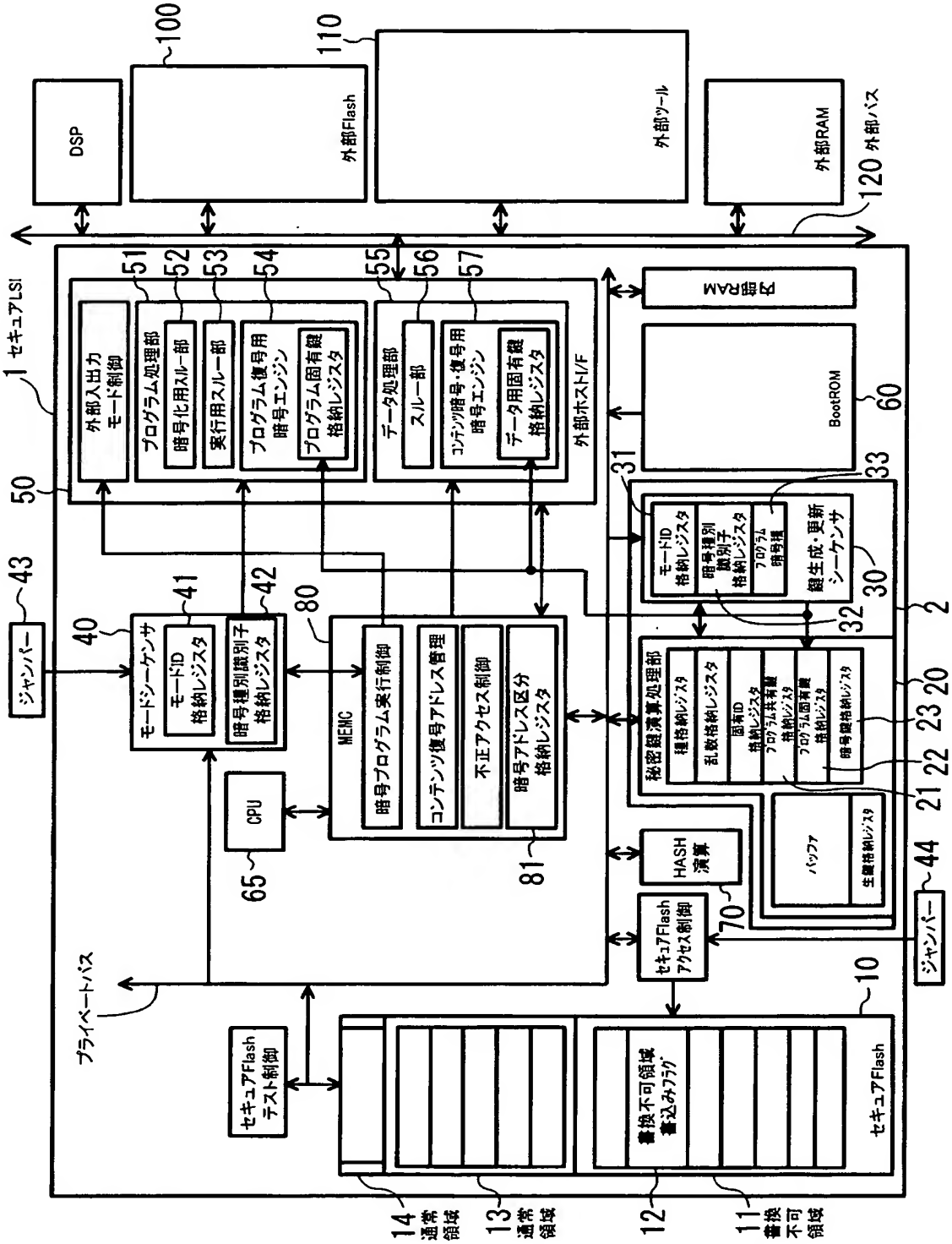
５８　プログラム固有鍵格納レジスタ

６０　ブートＲＯＭ

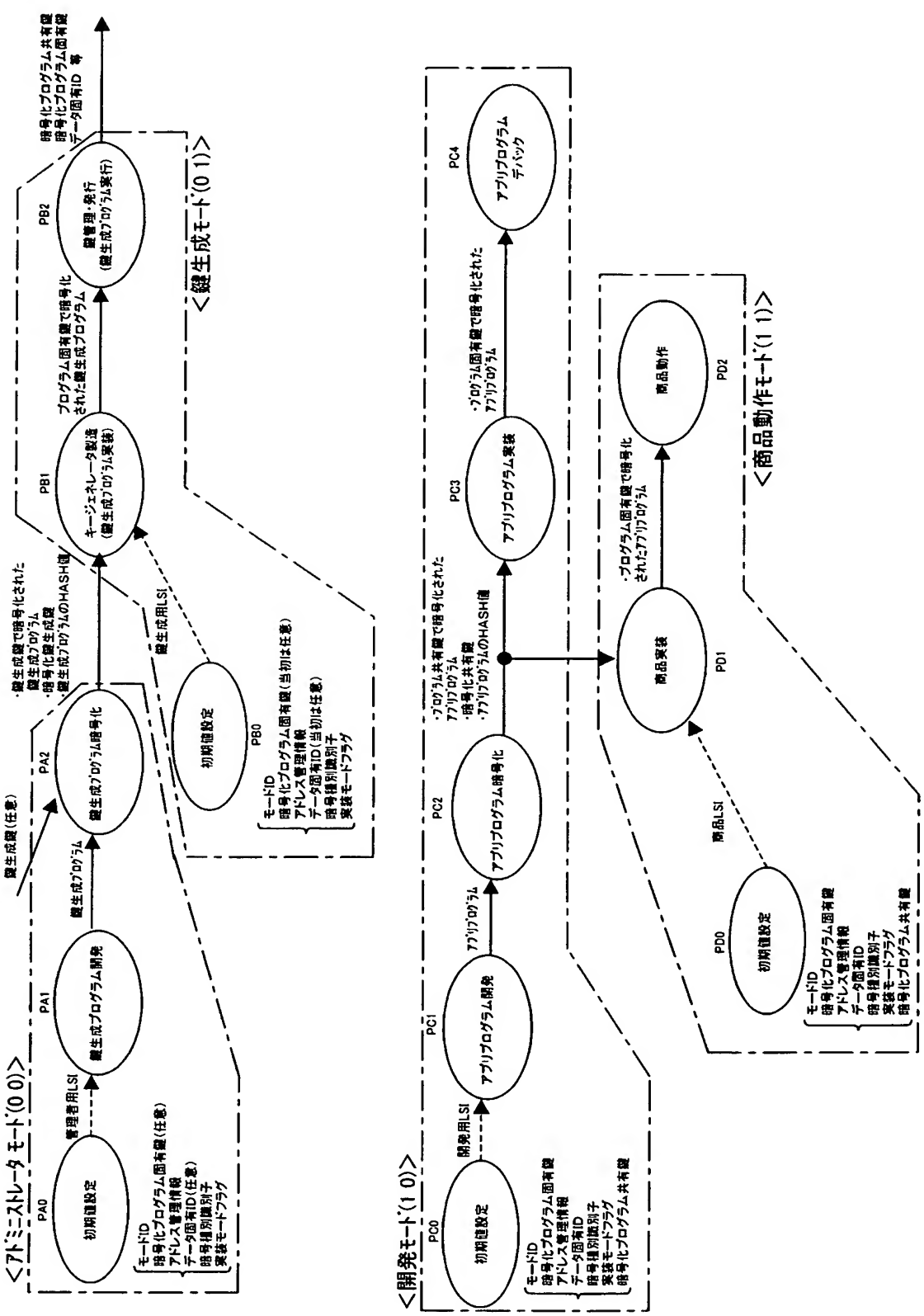
１００　外部メモリ

【書類名】　図面

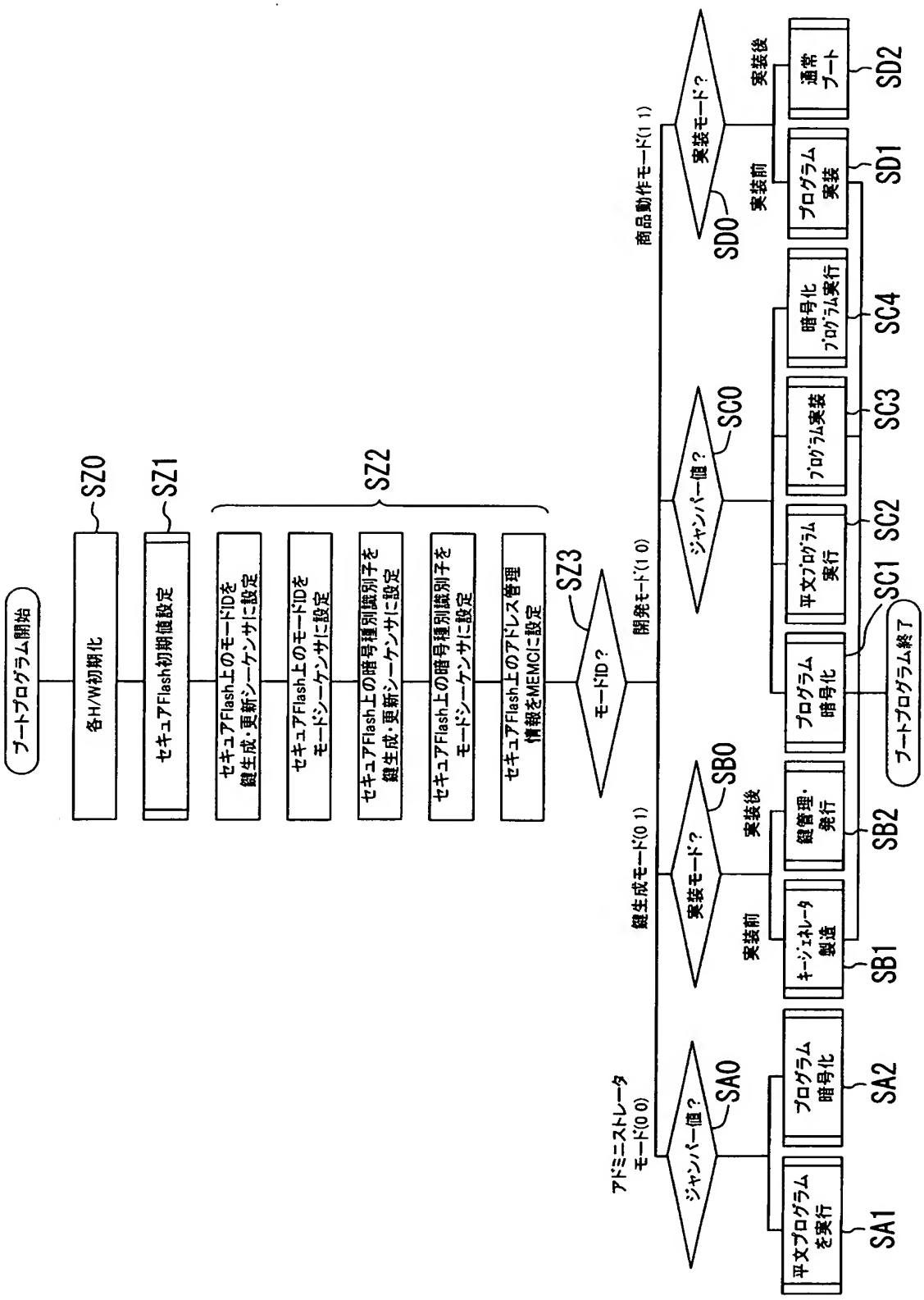
【図１】


DSP
外部Flash
外部ツール
外部RAM
外部バス
セキュアLSI
外部入出力モード制御
プログラム処理部
暗号化用スルー部
実行用スルー部
プログラム復号用暗号エンジン
プログラム固有鍵格納レジスタ
データ処理部
スルー部
コンテンツ暗号・復号用暗号エンジン
データ用固有鍵格納レジスタ
外部ホストI/F
内部RAM
BootROM
モードID格納レジスタ
暗号種別識別子格納レジスタ
プログラム暗号鍵
鍵生成・更新シーケンサ
ジャンパー
モードシーケンサ
モードID格納レジスタ
暗号種別識別子格納レジスタ
MEMC
暗号プログラム実行制御
コンテンツ復号アドレス管理
不正アクセス制御
暗号アドレス区分格納レジスタ
CPU
秘密鍵演算処理部
種格納レジスタ
乱数格納レジスタ
固有ID格納レジスタ
プログラム共有鍵格納レジスタ
プログラム固有鍵格納レジスタ
暗号鍵格納レジスタ
バッファ
生鍵格納レジスタ
HASH演算
セキュアFlashアクセス制御
ジャンパー
プライベートバス
セキュアFlashテスト制御
書換不可領域書込みフラグ
セキュアFlash
通常領域
通常領域
書換不可領域

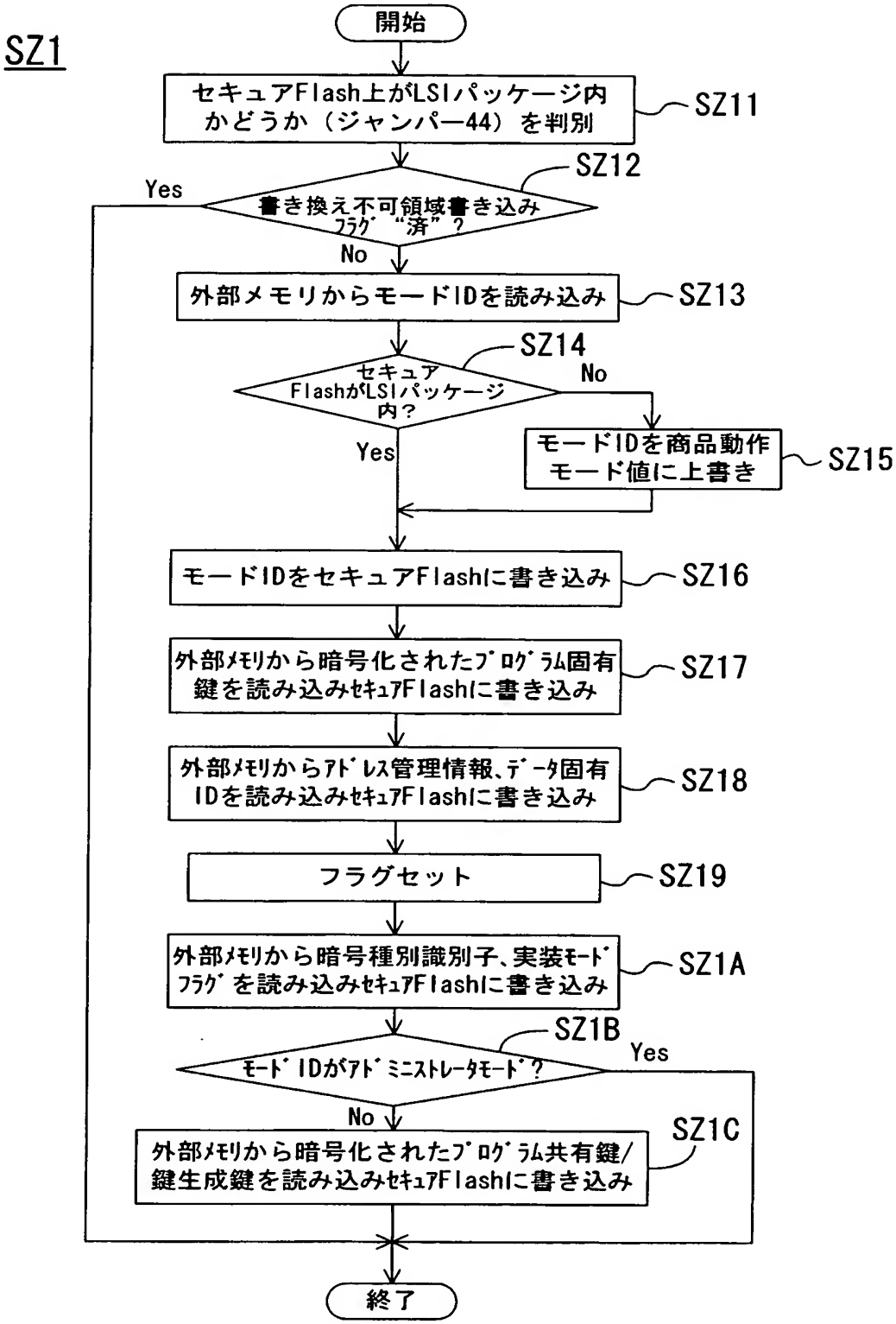
【図２】


<アドミニストレータモード(0 0)>
PA0
初期値設定
管理者用LSI
PA1
鍵生成プログラム開発
鍵生成プログラム
PA2
鍵生成プログラム暗号化
鍵生成鍵(任意)
・鍵生成鍵で暗号化された
鍵生成プログラム
・暗号化鍵生成プログラム
鍵生成プログラムのHASH値
PB1
キージェネレータ製造
(鍵生成プログラム実装)
プログラム固有鍵で暗号化
された鍵生成プログラム
PB2
鍵管理・発行
(鍵生成プログラム実行)
暗号化プログラム共有鍵
暗号化プログラム固有鍵
データ固有ID 等
<鍵生成モード(0 1)>
鍵生成用LSI
PB0
初期値設定
モードID
暗号化プログラム固有鍵(当初は任意)
アドレス管理情報
データ固有ID(当初は任意)
暗号種別識別子
実装モードフラグ
モードID
暗号化プログラム固有鍵(任意)
アドレス管理情報
データ固有ID(任意)
暗号種別識別子
実装モードフラグ
<開発モード(1 0)>
PC0
初期値設定
開発用LSI
PC1
アプリプログラム開発
アプリプログラム
PC2
アプリプログラム暗号化
・プログラム共有鍵で暗号化された
アプリプログラム
・暗号化共有鍵
・アプリプログラムのHASH値
PC3
アプリプログラム実装
・プログラム固有鍵で暗号化された
アプリプログラム
PC4
アプリプログラム
デバック
モードID
暗号化プログラム固有鍵
アドレス管理情報
データ固有ID
暗号種別識別子
実装モードフラグ
暗号化プログラム共有鍵
<商品動作モード(1 1)>
PD0
初期値設定
商品LSI
PD1
商品実装
・プログラム固有鍵で暗号化
されたアプリプログラム
PD2
商品動作

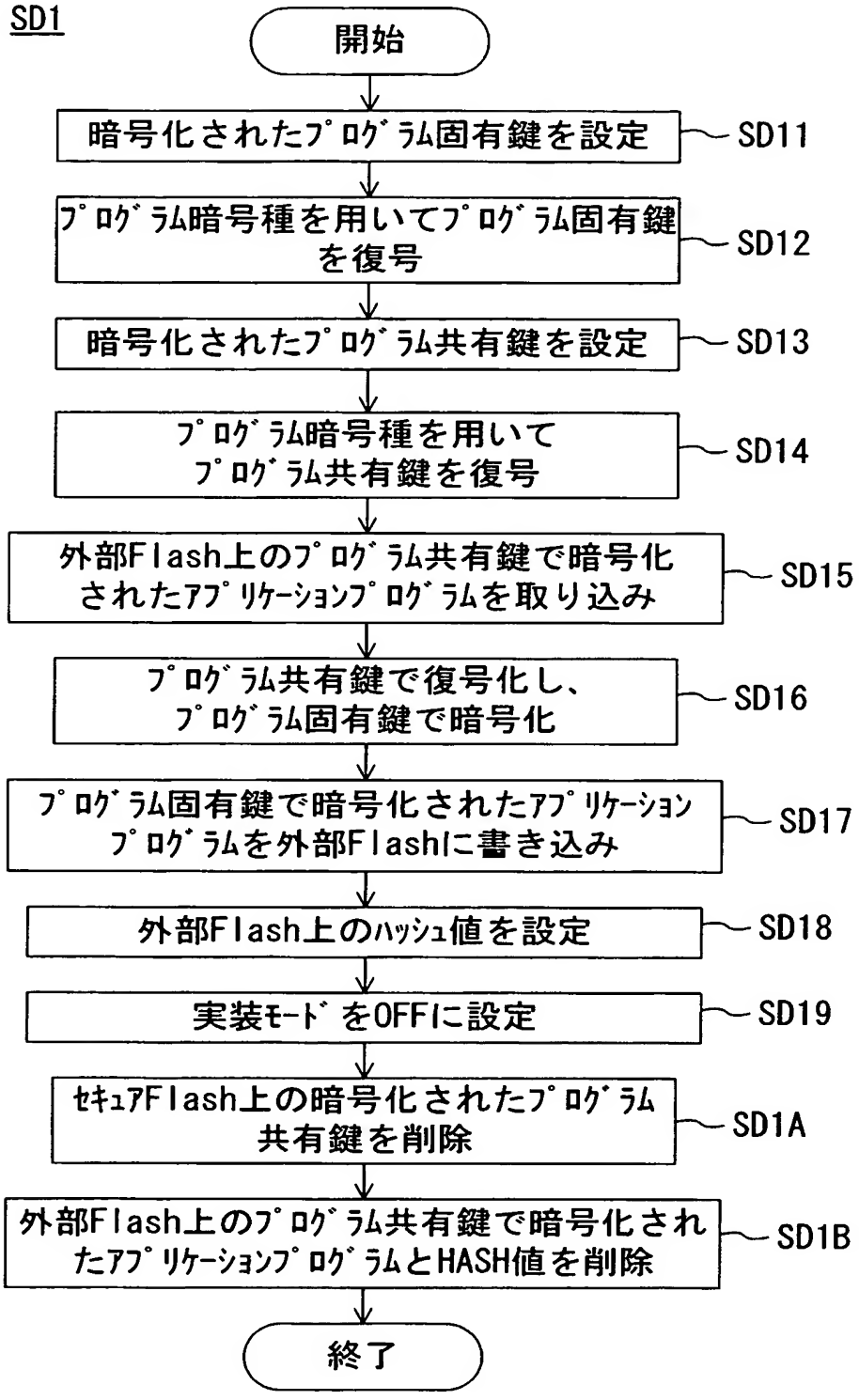
【図３】


ブートプログラム開始
SZ0
各H/W初期化
SZ1
セキュアFlash初期値設定
SZ2
セキュアFlash上のモードIDを鍵生成・更新シーケンサに設定
セキュアFlash上のモードIDをモードシーケンサに設定
セキュアFlash上の暗号種別識別子を鍵生成・更新シーケンサに設定
セキュアFlash上の暗号種別識別子をモードシーケンサに設定
セキュアFlash上のアドレス管理情報をMEMCに設定
SZ3
モードID？
アドミニストレータモード(00)
鍵生成モード(01)
開発モード(10)
商品動作モード(11)
ジャンパー値？
SA0
実装モード？
SB0
ジャンパー値？
SC0
実装モード？
SD0
平文プログラムを実行
SA1
プログラム暗号化
SA2
実装前
キージェネレータ製造
SB1
実装後
鍵管理・発行
SB2
プログラム暗号化
SC1
平文プログラム実行
SC2
プログラム実装
SC3
暗号化プログラム実行
SC4
ブートプログラム終了
実装前
プログラム実装
SD1
実装後
通常ブート
SD2

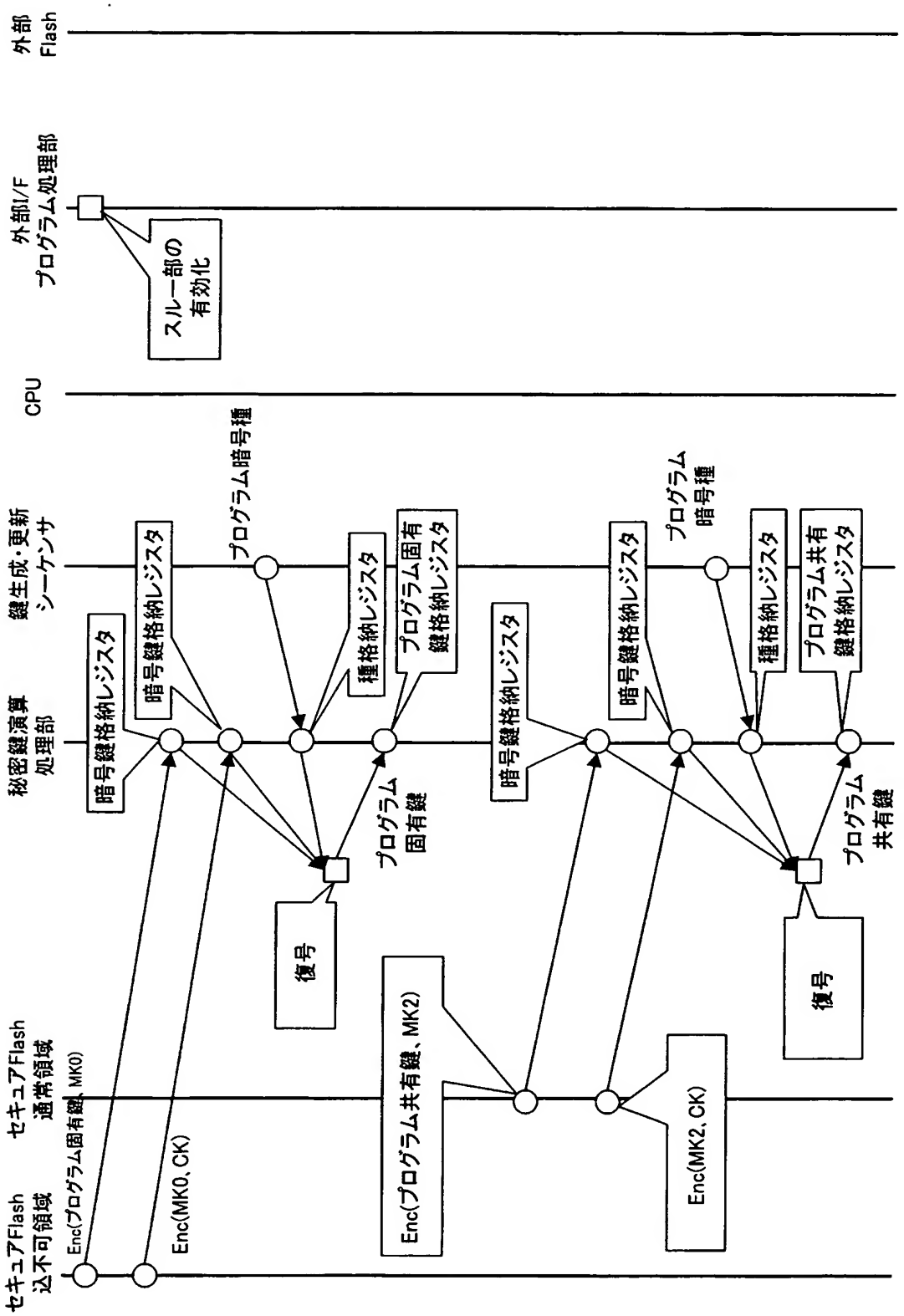
【図４】


SZ1
開始
セキュアFlash上がLSIパッケージ内かどうか（ジャンパー44）を判別
SZ11
SZ12
書き換え不可領域書き込みﾌﾗｸﾞ“済”？
Yes
No
外部メモリからモードIDを読み込み
SZ13
SZ14
セキュアFlashがLSIパッケージ内？
No
Yes
モードIDを商品動作モード値に上書き
SZ15
モードIDをセキュアFlashに書き込み
SZ16
外部ﾒﾓﾘから暗号化されたﾌﾟﾛｸﾞﾗﾑ固有鍵を読み込みｾｷｭｱFlashに書き込み
SZ17
外部ﾒﾓﾘからｱﾄﾞﾚｽ管理情報、ﾃﾞｰﾀ固有IDを読み込みｾｷｭｱFlashに書き込み
SZ18
フラグセット
SZ19
外部ﾒﾓﾘから暗号種別識別子、実装ﾓｰﾄﾞﾌﾗｸﾞを読み込みｾｷｭｱFlashに書き込み
SZ1A
SZ1B
ﾓｰﾄﾞIDがｱﾄﾞﾐﾆｽﾄﾚｰﾀﾓｰﾄﾞ？
Yes
No
外部ﾒﾓﾘから暗号化されたﾌﾟﾛｸﾞﾗﾑ共有鍵/鍵生成鍵を読み込みｾｷｭｱFlashに書き込み
SZ1C
終了

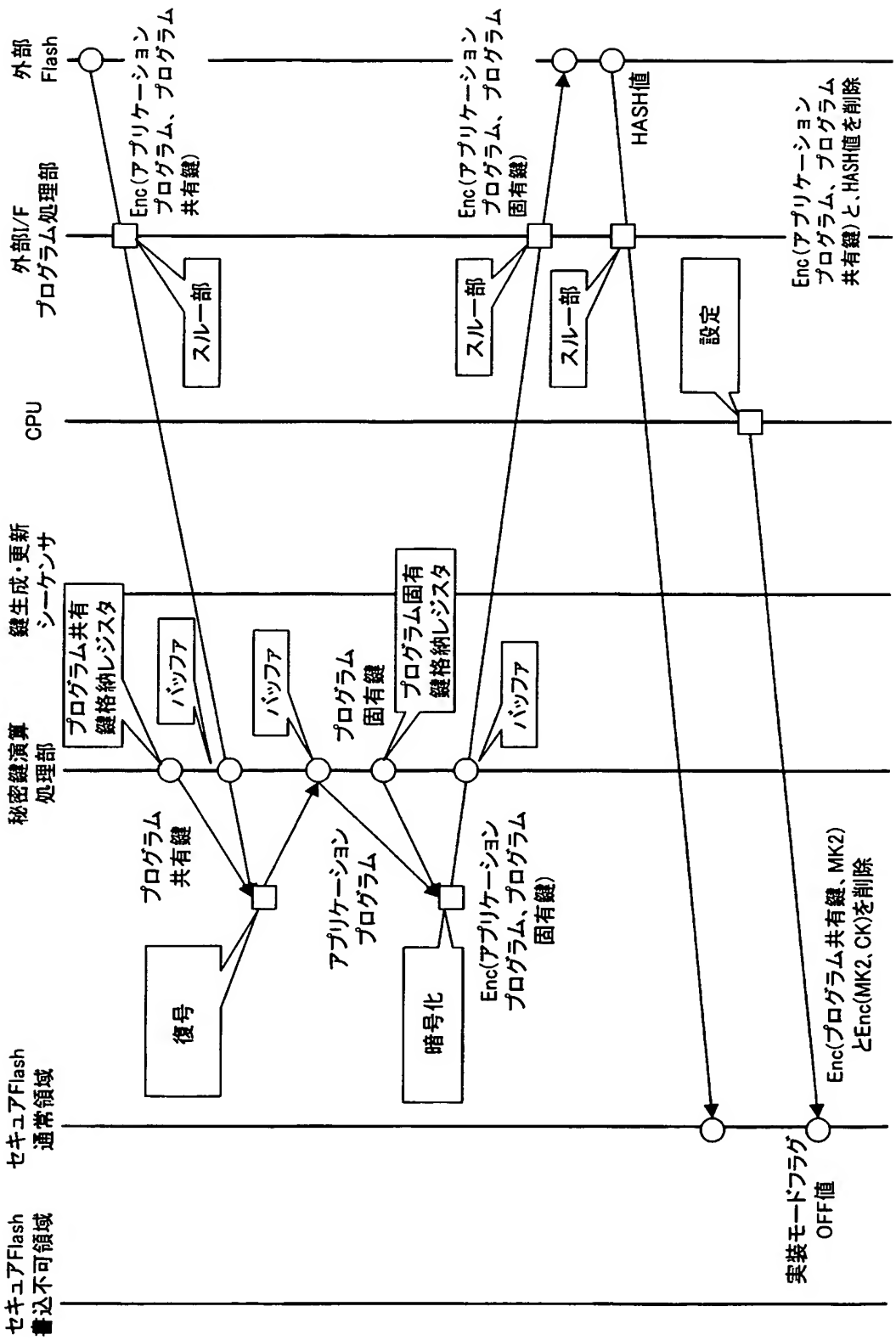
【図５】


SD1
開始
暗号化されたﾌﾟﾛｸﾞﾗﾑ固有鍵を設定
SD11
ﾌﾟﾛｸﾞﾗﾑ暗号種を用いてﾌﾟﾛｸﾞﾗﾑ固有鍵を復号
SD12
暗号化されたﾌﾟﾛｸﾞﾗﾑ共有鍵を設定
SD13
ﾌﾟﾛｸﾞﾗﾑ暗号種を用いてﾌﾟﾛｸﾞﾗﾑ共有鍵を復号
SD14
外部Flash上のﾌﾟﾛｸﾞﾗﾑ共有鍵で暗号化されたｱﾌﾟﾘｹｰｼｮﾝﾌﾟﾛｸﾞﾗﾑを取り込み
SD15
ﾌﾟﾛｸﾞﾗﾑ共有鍵で復号化し、ﾌﾟﾛｸﾞﾗﾑ固有鍵で暗号化
SD16
ﾌﾟﾛｸﾞﾗﾑ固有鍵で暗号化されたｱﾌﾟﾘｹｰｼｮﾝﾌﾟﾛｸﾞﾗﾑを外部Flashに書き込み
SD17
外部Flash上のﾊｯｼｭ値を設定
SD18
実装ﾓｰﾄﾞをOFFに設定
SD19
ｾｷｭｱFlash上の暗号化されたﾌﾟﾛｸﾞﾗﾑ共有鍵を削除
SD1A
外部Flash上のﾌﾟﾛｸﾞﾗﾑ共有鍵で暗号化されたｱﾌﾟﾘｹｰｼｮﾝﾌﾟﾛｸﾞﾗﾑとHASH値を削除
SD1B
終了

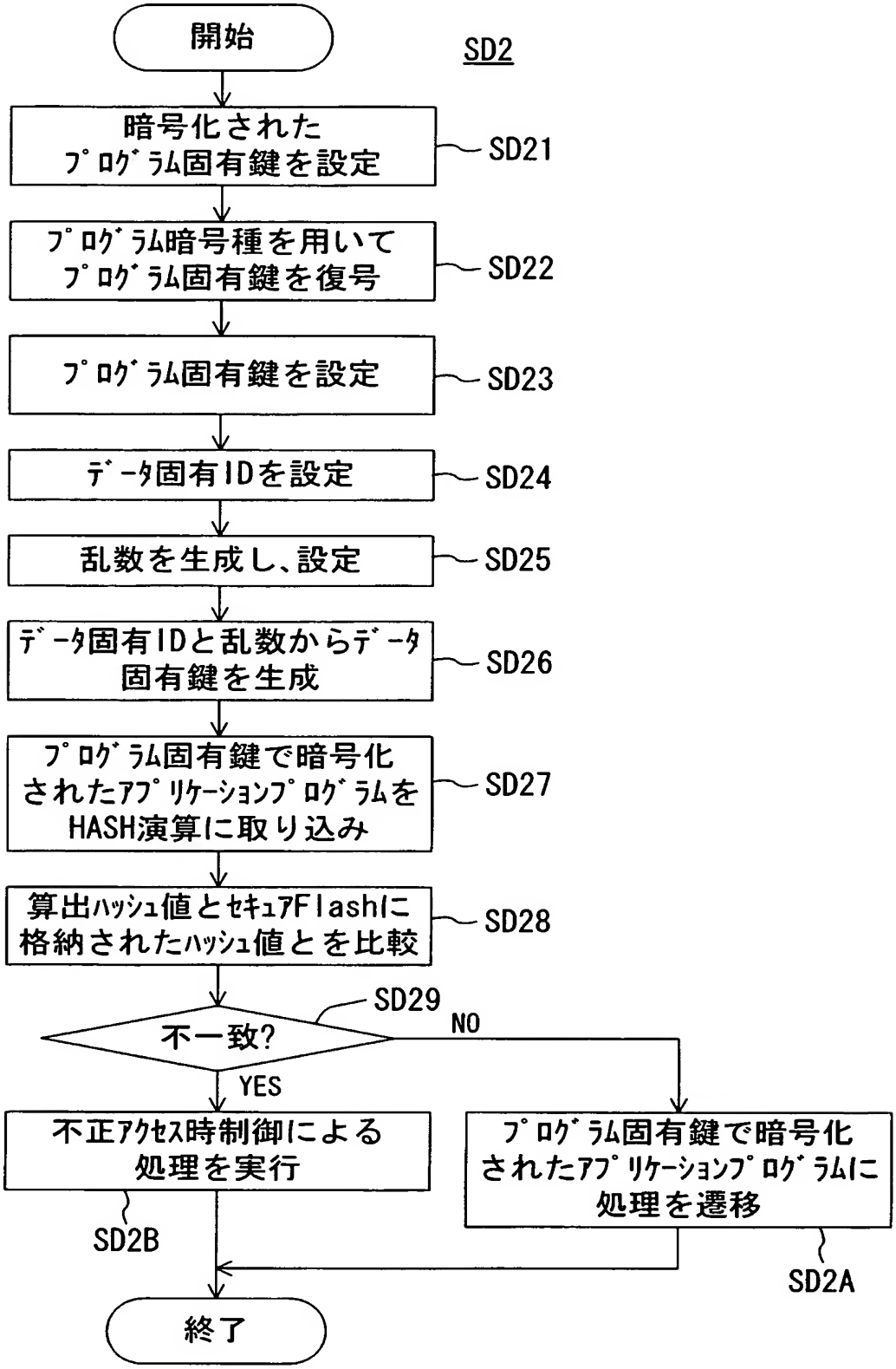
【図６】


外部 Flash
外部I/F プログラム処理部
スルー部の 有効化
CPU
鍵生成・更新 シーケンサ
プログラム暗号種
プログラム 暗号種
秘密鍵演算 処理部
暗号鍵格納レジスタ
暗号鍵格納レジスタ
種格納レジスタ
プログラム固有 鍵格納レジスタ
暗号鍵格納レジスタ
暗号鍵格納レジスタ
種格納レジスタ
プログラム共有 鍵格納レジスタ
プログラム 固有鍵
プログラム 共有鍵
復号
復号
セキュアFlash 込不可領域
セキュアFlash 通常領域
Enc(プログラム固有鍵, MK0)
Enc(MK0, CK)
Enc(プログラム共有鍵, MK2)
Enc(MK2, CK)

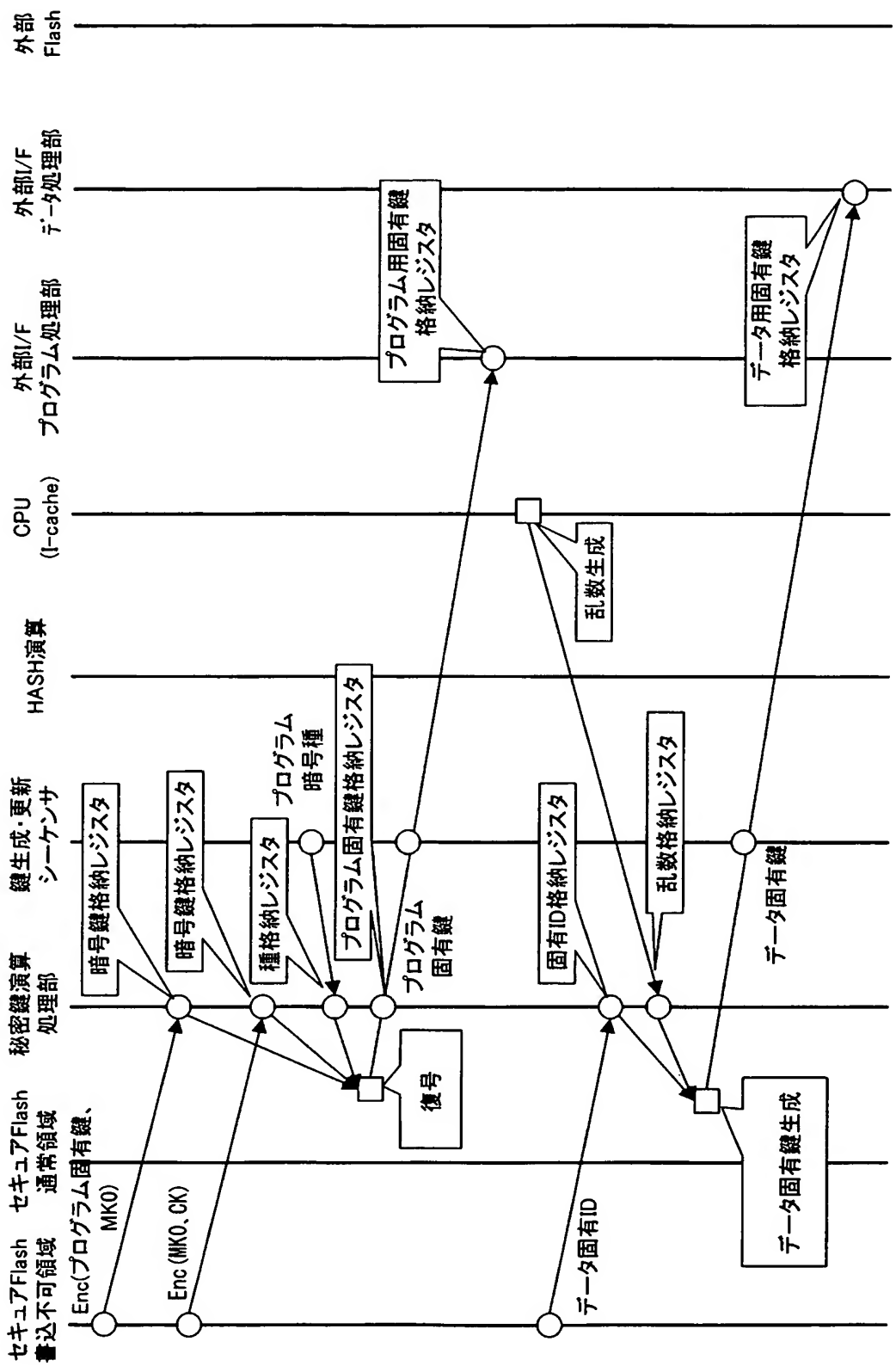
【図７】


外部 Flash
外部I/F プログラム処理部
CPU
鍵生成・更新 シーケンサ
秘密鍵演算 処理部
セキュアFlash 通常領域
セキュアFlash 書込不可領域
Enc(アプリケーション プログラム、プログラム 共有鍵)
スルー部
Enc(アプリケーション プログラム、プログラム 固有鍵)
スルー部
HASH値
スルー部
設定
Enc(アプリケーション プログラム、プログラム 共有鍵)と、HASH値を削除
プログラム共有 鍵格納レジスタ
バッファ
バッファ
プログラム 固有鍵
プログラム固有 鍵格納レジスタ
バッファ
プログラム 共有鍵
アプリケーション プログラム
Enc(アプリケーション プログラム、プログラム 固有鍵)
復号
暗号化
Enc(プログラム共有鍵、MK2) とEnc(MK2、CK)を削除
実装モードフラグ OFF値

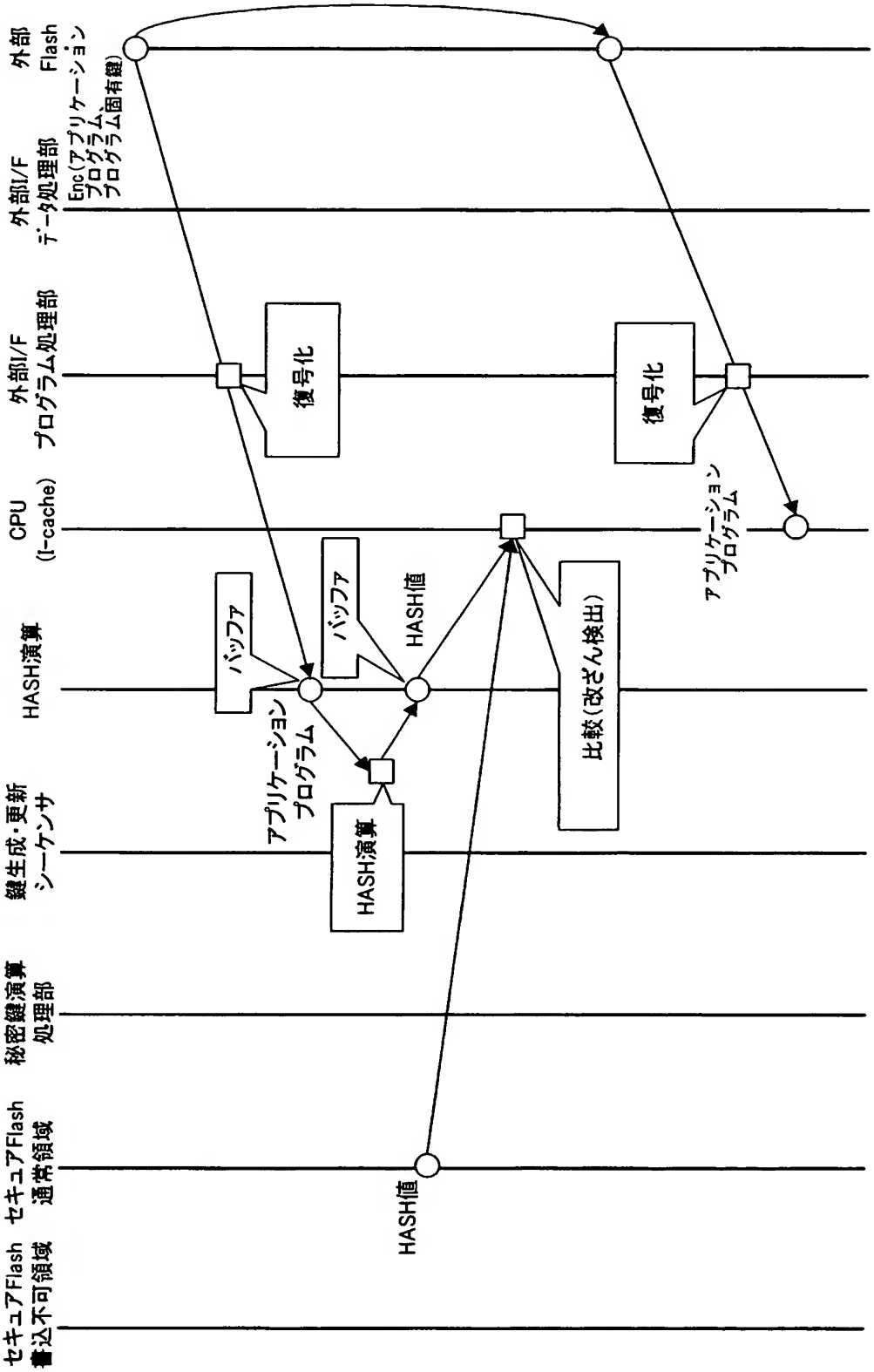
【図８】


開始
SD2
暗号化された
ﾌﾟﾛｸﾞﾗﾑ固有鍵を設定
SD21
ﾌﾟﾛｸﾞﾗﾑ暗号種を用いて
ﾌﾟﾛｸﾞﾗﾑ固有鍵を復号
SD22
ﾌﾟﾛｸﾞﾗﾑ固有鍵を設定
SD23
ﾃﾞｰﾀ固有IDを設定
SD24
乱数を生成し、設定
SD25
ﾃﾞｰﾀ固有IDと乱数からﾃﾞｰﾀ
固有鍵を生成
SD26
ﾌﾟﾛｸﾞﾗﾑ固有鍵で暗号化
されたｱﾌﾟﾘｹｰｼｮﾝﾌﾟﾛｸﾞﾗﾑを
HASH演算に取り込み
SD27
算出ﾊｯｼｭ値とｾｷｭｱFlashに
格納されたﾊｯｼｭ値とを比較
SD28
SD29
不一致?
NO
YES
不正ｱｸｾｽ時制御による
処理を実行
SD2B
ﾌﾟﾛｸﾞﾗﾑ固有鍵で暗号化
されたｱﾌﾟﾘｹｰｼｮﾝﾌﾟﾛｸﾞﾗﾑに
処理を遷移
SD2A
終了

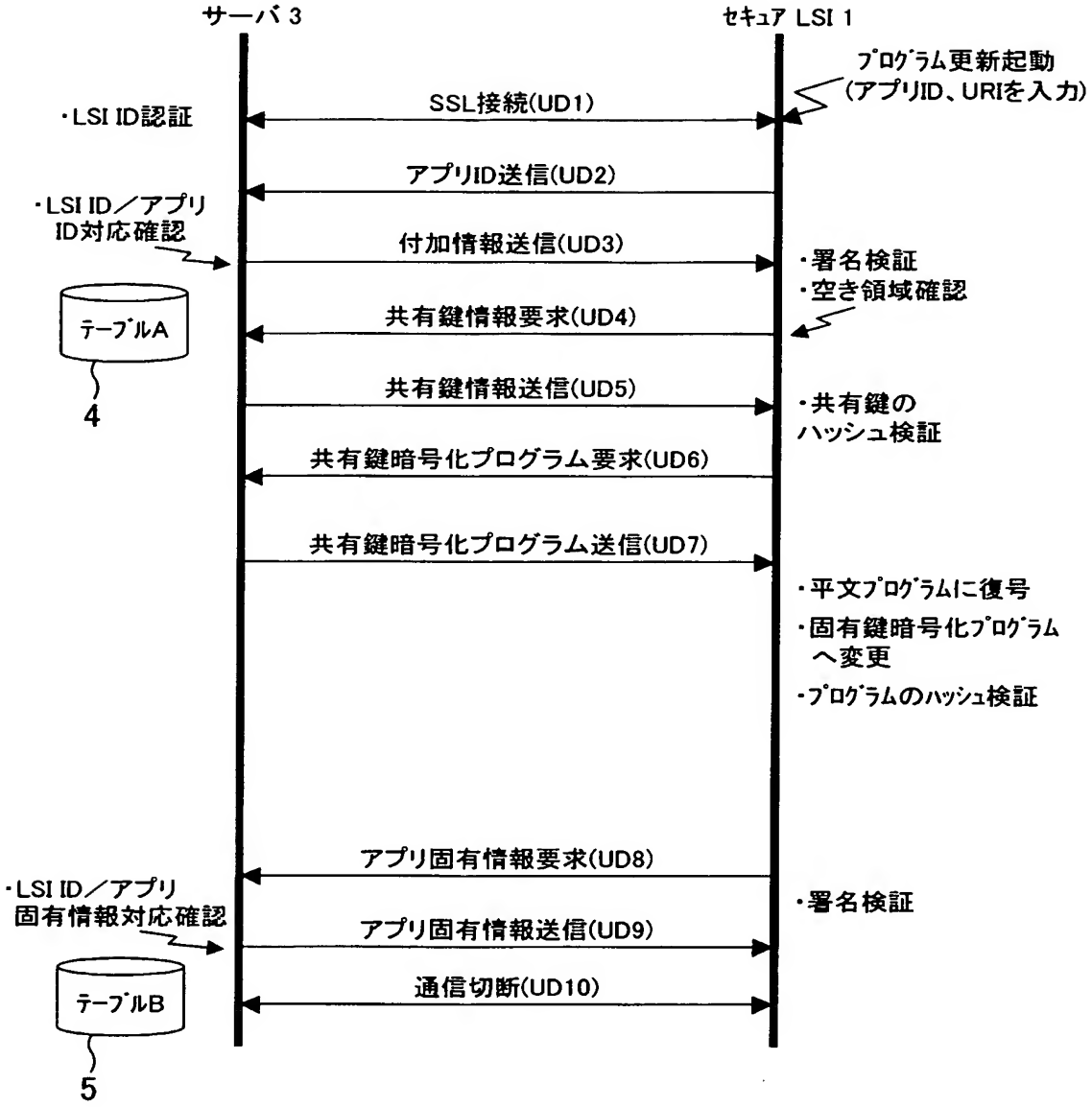
【図９】


セキュアFlash 書込不可領域
セキュアFlash 通常領域
秘密鍵演算 処理部
鍵生成・更新 シーケンサ
HASH演算
CPU (I-cache)
外部I/F プログラム処理部
外部I/F データ処理部
外部 Flash
Enc(プログラム固有鍵、MK0)
Enc (MK0、CK)
暗号鍵格納レジスタ
暗号鍵格納レジスタ
種格納レジスタ
プログラム 暗号種
プログラム固有鍵格納レジスタ
プログラム 固有鍵
復号
プログラム用固有鍵 格納レジスタ
乱数生成
固有ID格納レジスタ
乱数格納レジスタ
データ固有鍵
データ用固有鍵 格納レジスタ
データ固有ID
データ固有鍵生成

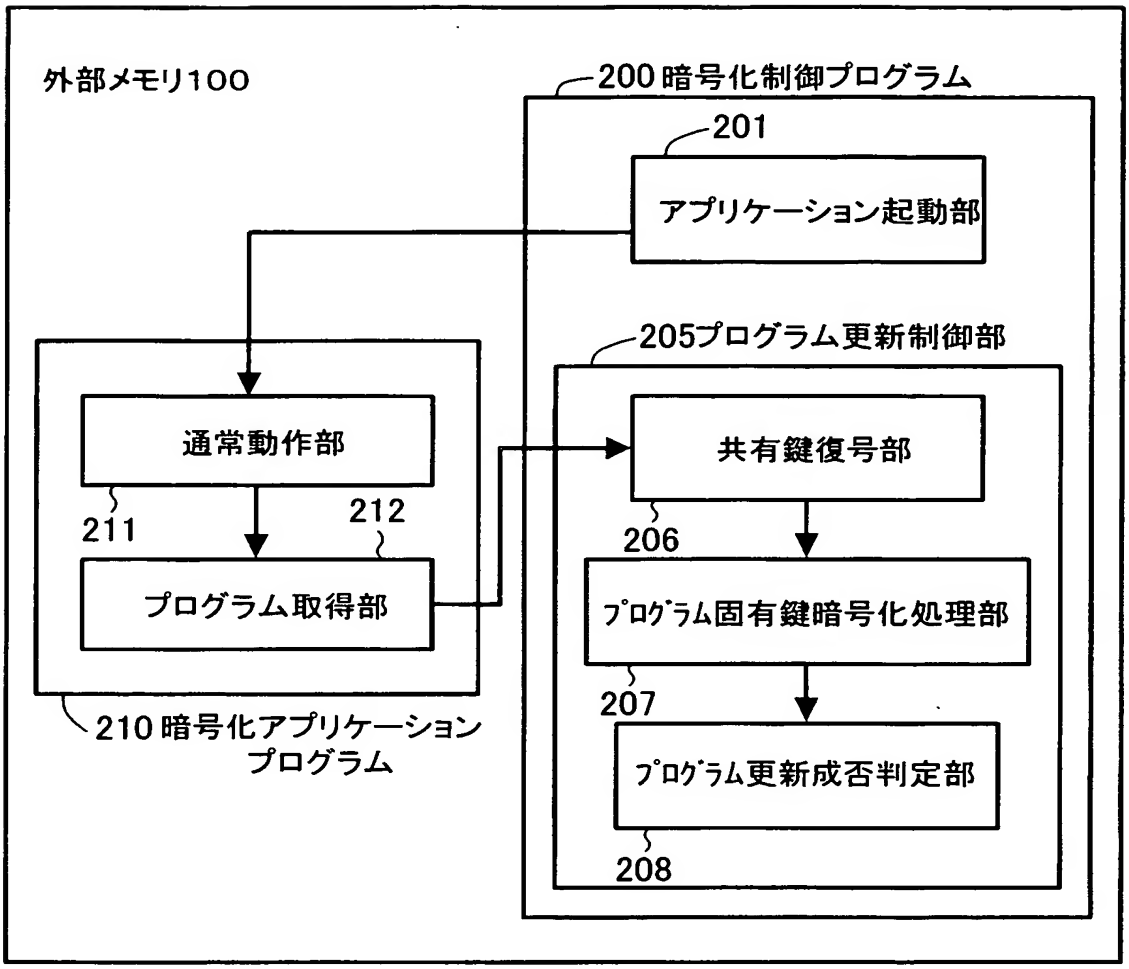
【図１０】


外部 Flash
外部I/F データ処理部
Enc(アプリケーション プログラム固有鍵)
外部I/F プログラム処理部
復号化
復号化
CPU (I-cache)
アプリケーション プログラム
HASH演算
バッファ
バッファ
HASH値
アプリケーション プログラム
比較(改ざん検出)
鍵生成・更新 シーケンサ
HASH演算
秘密鍵演算 処理部
セキュアFlash 通常領域
HASH値
セキュアFlash 書込不可領域

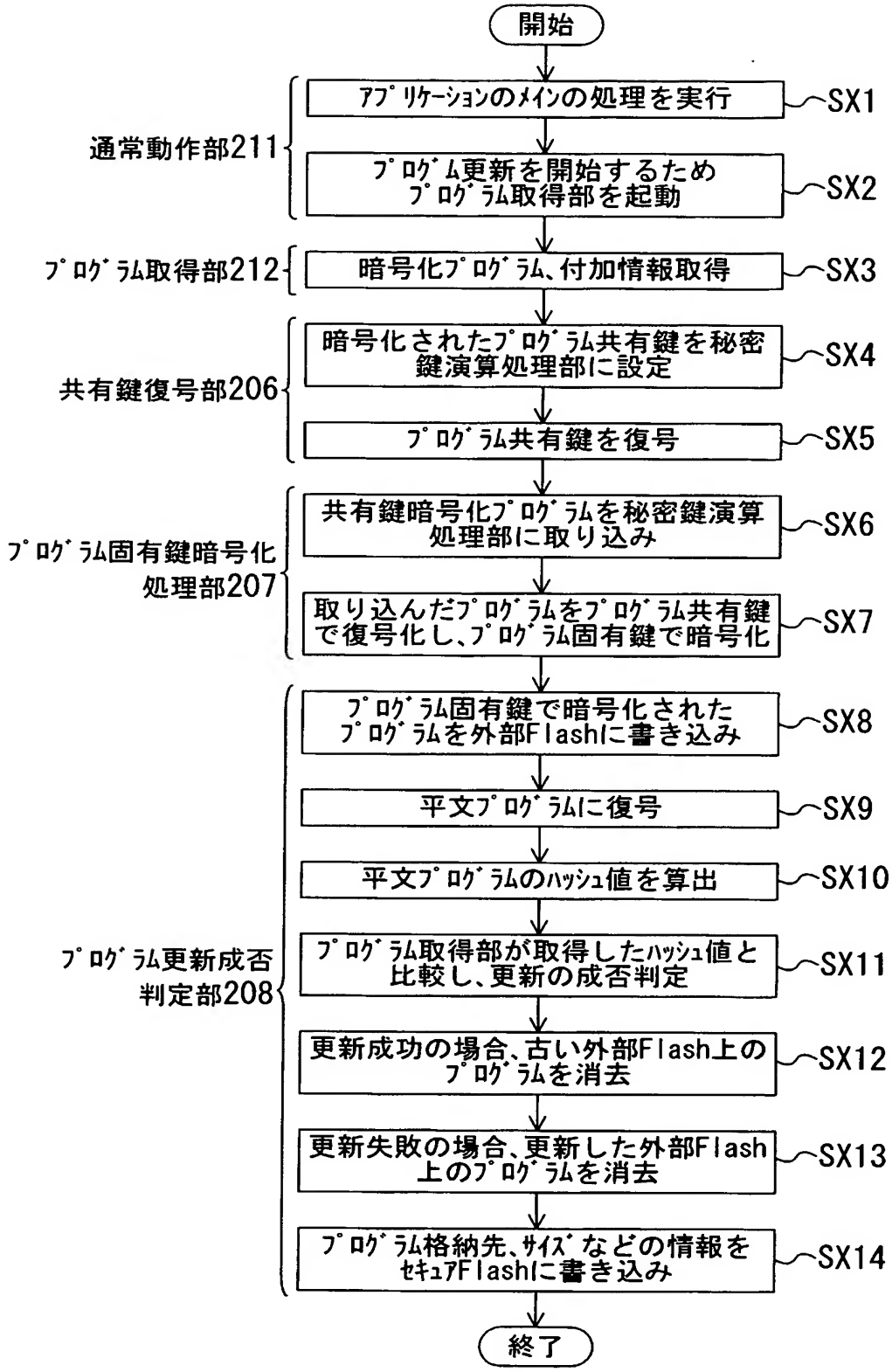
【図１１】


サーバ 3
セキュア LSI 1
プログラム更新起動
(アプリID、URIを入力)
・LSI ID認証
SSL接続(UD1)
アプリID送信(UD2)
・LSI ID／アプリ
ID対応確認
付加情報送信(UD3)
・署名検証
・空き領域確認
テーブルA
共有鍵情報要求(UD4)
4
共有鍵情報送信(UD5)
・共有鍵の
ハッシュ検証
共有鍵暗号化プログラム要求(UD6)
共有鍵暗号化プログラム送信(UD7)
・平文プログラムに復号
・固有鍵暗号化プログラム
へ変更
・プログラムのハッシュ検証
アプリ固有情報要求(UD8)
・LSI ID／アプリ
固有情報対応確認
・署名検証
アプリ固有情報送信(UD9)
テーブルB
通信切断(UD10)
5

【図１２】


外部メモリ100
200 暗号化制御プログラム
201
アプリケーション起動部
205プログラム更新制御部
通常動作部
共有鍵復号部
211
212
206
プログラム取得部
ﾌﾟﾛｸﾞﾗﾑ固有鍵暗号化処理部
207
ﾌﾟﾛｸﾞﾗﾑ更新成否判定部
208
210 暗号化アプリケーション
プログラム

【図１３】


開始
通常動作部211
ｱﾌﾟﾘｹｰｼｮﾝのﾒｲﾝの処理を実行
SX1
ﾌﾟﾛｸﾞﾗﾑ更新を開始するため
ﾌﾟﾛｸﾞﾗﾑ取得部を起動
SX2
ﾌﾟﾛｸﾞﾗﾑ取得部212
暗号化ﾌﾟﾛｸﾞﾗﾑ、付加情報取得
SX3
共有鍵復号部206
暗号化されたﾌﾟﾛｸﾞﾗﾑ共有鍵を秘密
鍵演算処理部に設定
SX4
ﾌﾟﾛｸﾞﾗﾑ共有鍵を復号
SX5
ﾌﾟﾛｸﾞﾗﾑ固有鍵暗号化
処理部207
共有鍵暗号化ﾌﾟﾛｸﾞﾗﾑを秘密鍵演算
処理部に取り込み
SX6
取り込んだﾌﾟﾛｸﾞﾗﾑをﾌﾟﾛｸﾞﾗﾑ共有鍵
で復号化し、ﾌﾟﾛｸﾞﾗﾑ固有鍵で暗号化
SX7
ﾌﾟﾛｸﾞﾗﾑ更新成否
判定部208
ﾌﾟﾛｸﾞﾗﾑ固有鍵で暗号化された
ﾌﾟﾛｸﾞﾗﾑを外部Flashに書き込み
SX8
平文ﾌﾟﾛｸﾞﾗﾑに復号
SX9
平文ﾌﾟﾛｸﾞﾗﾑのﾊｯｼｭ値を算出
SX10
ﾌﾟﾛｸﾞﾗﾑ取得部が取得したﾊｯｼｭ値と
比較し、更新の成否判定
SX11
更新成功の場合、古い外部Flash上の
ﾌﾟﾛｸﾞﾗﾑを消去
SX12
更新失敗の場合、更新した外部Flash
上のﾌﾟﾛｸﾞﾗﾑを消去
SX13
ﾌﾟﾛｸﾞﾗﾑ格納先、ｻｲｽﾞなどの情報を
ｾｷｭｱFlashに書き込み
SX14
終了

【書類名】　要約書

【要約】

【課題】　固有鍵で暗号化されたプログラムを実行可能なＬＳＩについて、高いセキュリティを保ちつつ、プログラムを更新可能にする。

【解決手段】　セキュアＬＳＩ１を含むシステムは、サーバ３との通信路を確立し（ＵＤ１）、サーバ３から送信された，共有鍵で暗号化された共有鍵暗号化プログラムを受信する（ＵＤ６，ＵＤ７）。そして、受信した共有鍵暗号化プログラムを復号して平文プログラムを生成し、さらに、平文プログラムを固有鍵で再暗号化し、新たな固有鍵暗号化プログラムとして外部メモリに格納する。

【選択図】　図１１

特願２００２－３３１９９２

出　願　人　履　歴　情　報

識別番号　　［０００００５８２１］

１．変更年月日　　１９９０年　８月２８日
［変更理由］　　新規登録
住　所　　大阪府門真市大字門真１００６番地
氏　名　　松下電器産業株式会社